40. 연계 요구사항 분석

1. 통합 구현의 개념 – 사용자의 요구사항에 맞춰 송 수신 모듈과 중계 모듈 간의 연계를 구현

2. 송신 시스템과 모듈 – 데이터를 생성 및 변환하여 전송하는 시스템

송신 모듈 : 전송 데이터를 생성, 필요에 따라 전송 데이터의 변환 등을 수행

모니터링 기능 : 전송 데이터의 생성부터 송신까지의 과정과 송신 상태 등을 확인

3. 수신 시스템과 모듈 – 수신 받은 데이터를 정제 및 변환하는 시스템

수신 모듈 : 수신 데이터를 정제하고 애플리케이션이나 DB 테이블에 적합한 데이터로 변환하는 작업 수행

모니터링 기능 : 수신 모듈과 연계 데이터의 수신 상태, 오류 처리, 데이터 반영 등을 확인

4. 중계 시스템 – 내 외부 시스템 간 또는 내부 시스템 간의 연계 시 사용되는 아키텍처

송신 데이터의 오류 처리, 수신 시스템에 맞는 데이터 형식으로의 변환 등을 수행

5. 연계 데이터 – 송 수신 시스템 간 송 수신되는 데이터, 속성/길이/타입, DB테이블/DB 파일 구분

6. 네트워크 – 송신 시스템, 수신 시스템, 중계 시스템을 연결해주는 통신망, 공중망/사설망/프로토콜 포함

7. 연계 요구사항 분석 – 통합 구현을 위해 사용자 요구사항을 분석하고 연계 데이터를 식별 및 표준화

연계 요구사항 분석 절차

1. 시스템 구성도, 응용 앱 구성 등을 통해 송 수신 시스템의 하드웨어 및 소프트웨어 구성, 네트워크 현황 확인

2. 송 수신 시스템과 연결할 데이터와 관련된 테이블 정의서, 코드 정의서 등의 문서를 확인

3. 확인된 시스템의 구성과 데이터 현황 정보 등을 기반으로 체크리스트를 작성

4. 시스템 구성도, 응용 앱 구성, 테이블 정의서, 코드 정의서, 체크 리스트 등을 공유하고 인터뷰, 설문조사 실시

5. 조사로 확인된 결과를 기반으로 요구사항의 ID, 이름, 유형, 설명, 해결방안 등이 포함된 요구사항 분석서 정의

41. 연계 데이터 식별 및 표준화

연계 데이터를 구성하기 위해 연계 데이터를 식별하고 식별된 연계 데이터를 표준화한 후 연계 정의서 작성

연계 데이터 식별 및 표준화 절차

1. 연계 범위 및 항목 정의 – 연계 범위와 항목은 시스템 간 연계할 정보를 상세화하여 정의

2. 연계 코드 변환 및 매핑 – 연계 대상 범위 및 항목 정의 후 정보로 전환하거나 검색 조건으로 활용하기 위해 연계 정보 중 코드로 관리되어야 할 항목을 찾아 코드로 변호나

3. 연계 데이터 식별자와 변경 구분 추가 – 수신 시스템의 연계 정보에 송신 시스템의 식별자 추가

4. 연계 데이터 표현 방법 정의 – 연계 대상 범위 및 항목과 코드 변환 및 매핑 방식 정의 후 연계 데이터 구성

연계 데이터 형식 : DB테이블, DB파일로 구분, DB파일은 태그나 , : ; 에 의해 구분 가능

5. 연계 정의서 및 명세서 작성 – 연계 항목과 데이터의 타입, 길이 등을 구성 후 형식을 정의한 결과 기반 작성

연계 정의서 : 송 수신 시스템의 시스템명, 인터페이스ID, 인터페이스 방식 및 주기 등 인터페이스 현황 기재

연계 명세서 : 연계 정의서에 작성한 인터페이스ID 별로 인터페이스 주기 및 방식, DB 및 파일 형식, 송 수신 시스템의 타입, 길이, 식별자, 코드화 여부 등의 세부 사항 작성

42. 연계 메커니즘 정의

1. 연계 메커니즘의 개요 – 데이터의 생성/전송하는 송신 체계, 수신 및 운영DB반영을 담당하는 수신 체계

송신 시스템은 운영 DB로부터 인터페이스 테이블이나 파일(xml, text..) 형식으로 연계 데이터를 생성하여 송신

수신 시스템은 송신 시스템으로부터 전송된 데이터를 받아 수신 시스템에 맞는 데이터로 변환 후 운영 DB 반영

송 수신 시스템 간 데이터의 송 수신과 송 수신 시스템 현황을 모니터링하는 중계 시스템을 설치 가능

시스템 간 역할 중복 방지를 위해 아키텍처 설계 후 인터페이스 테스트와 통합 테스트를 통해 기능 검증

2. 직접 연계 방식 – 중간 매개체 없이 송 수신 시스템이 직접 연계하는 방식

장점 : 연계 및 통합구현이 단순하고 용이, 개발 비용 저렴, 개발 기간 짧음, 데이터 연계 처리 성능 뛰어남

단점 : 송 수신 시스템간 결합도가 높아 시스템 변경 시 오류가 발생할 수 있음, 보안을 위한 암호화, 복호화 처리/비즈니스 로직 적용의 어려움, 연계 및 통합이 가능한 시스템 환경이 제한적

DB Link : DB에서 제공하는 DB Link 객체를 이용하는 방식

API(application Programming Interface)/Open API : 데이터를 송신 시스템의 DB에서 읽어와 제공, app인터페이스

**DB Connection** : 수신 시스템의 WAS에서 송신 시스템의 DB로 연결해주는 방식

**JDBC** : Java에서 DB에 접근하여 데이터를 삽입, 삭제, 수정, 조회할 수 있도록 java와 DB를 연결해 주는 방식

3. 간접 연계 방식 – 송 수신 시스템 사이에 중간 매개체를 두어 연계하는 방식

장점 : 서로 다른 네트워크, 프로토콜 등 다양한 환경의 연계 및 통합 가능, 보안 품질 보장과 비즈니스 처리를 위한 로직을 쉽게 반영, 송 수신 시스템간 인터페이스가 변경되어도 오류 없이 서비스 가능

단점 : 연계 메커니즘과 아키텍처 복잡, 중간 매개체로 인해 성능 저하, 개발 및 적용 기간이 비교적 김

종류

연계 솔루션 : EAI 서버와 송 수신 시스템에 설치되는 클라이언트를 이용하는 방식

ESB(Enterprise Service Bus) : app간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스 제공 방식

Socket : 서버는 통신을 위한 소켓을 생성 후 포트 할당, Client의 통신 요청 시 연결하여 통신하는 네트워크 기술

Web Service : 웹 서비스에서 WSDL과 UDDI, SOAP 프로토콜을 이용하여 연계하는 방식

4. 연계 메커니즘의 구성

**연계 데이터 생성 및 추출** – 연계 솔루션과 관계없이 응용 프로그램이나 DB등 응용 시스템에서 연계 데이터를 생성하고 추출, 오류 발생 시 오류 발생 시점, 오류 코드, 오류 내용 등을 로그 테이블이나 파일에 기록

**코드 매핑 및 데이터 변환** – 송신 시스템에서 사용하는 코드를 수신 시스템에서 사용하는 코드로 매핑 및 전환

**인터페이스 테이블 또는 파일 생성** – 연계 데이터를 인터페이스 테이블이나 파일 형식으로 생성하는 것

송신 시스템에서 데이터 생성 시 보안이 필요한 데이터는 암호화(Encryption), 수신 시스템에서 암호화된 데이터를 운영 DB에 반영 시 복호화(Decryption), 발생한 오류 내역은 로그 테이블이나 파일에 기록

**로그(Log) 기록** – 송 수신 시스템에서 수행되는 모든 과정에 대한 결과 및 오류에 대한 정보를 로그 테이블이나 파일에 기록, 모니터링 목적에 따라 로그 테이블과 파일의 관리 항목을 서로 다르게 기록할 수 있음

**연계 서버 또는 송 수신 어댑터** – 송 수신 시스템 중 한곳에 설치, 인터페이스 테이블 또는 파일의 데이터를 전송 형식에 맞게 변환하고 송 수신을 수행하는 등 송 수신과 관련된 모든 처리를 수행

송신 어댑터 : 송신 시스템에 설치, 데이터를 전송 형식에 맞도록 변환 후 송신을 수행

수신 어댑터 : 수신 시스템에 설치, 송신 시스템으로부터 수신한 데이터를 인터페이스 테이블이나 파일로 생성

전송 – 송신 시스템에서 생성된 연계 데이터를 네트워크 환경에 맞는 데이터로 변환, 수신 시스템으로 보냄

운영 DB에 연계 데이터 반영 – 데이터를 변환 프로그램을 이용해 수신 시스템의 운영 DB에 반영하는 것

수신된 데이터가 송신 시스템의 운영 DB에서 사용하는 공통 코드와 데이터 형식으로 전환, 확인하는 작업 포함

43. 연계 장애 및 오류처리 구현

1. 연계 메커니즘 구간별 장애 및 오류 모니터링 현황

송신 시스템

오류 발생 시점 : 데이터 생성 및 추출, 코드 매핑 및 데이터 변환, 인터페이스 테이블 또는 파일등록

오류 로그 기록 장소 : 송신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블

오류 로그 기록 주체 : 별도로 구현한 송신용 연계 프로그램

수신 시스템

오류 발생 시점 : 연계 데이터 로드, 코드 매핑 및 데이터 변환, 운영 DB에 반영

오류 로그 기록 장소 : 수신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블

오류 로그 기록 주체 : 별도로 구현한 수신용 연계 프로그램

연계 서버

오류 발생 시점 : 연계 데이터 로드 및 전송 형식으로 변환/송 수신, 수신 시스템의 데이터 형식 변환 및 로드

오류 로그 기록 장소 : 연계 서버에 설정된 로그 및 오류 로그 테이블 또는 파일

오류 로그 기록 주체 : 연계 서버

2. 장애 및 오류 유형과 처리 방안

송신 시스템의 연계 프로그램 오류 : 데이터 생성 및 추출 권한X, **연계 프로그램 구현상의 오류(아래 동일)**

수신 시스템의 연계 프로그램 : 운영 DB 접근 권한X, 데이터 변환 및 반영 시 예외 미 처리 ex)데이터 존재x

처리 방안 : 송 수신 연계 프로그램의 로그를 확인하여 원인 분석 후 처리, 데이터 전송이나 반영을 재작업

연계서버 오류 : 연계 서버 실행 송 수신, 전송 형식 변환 등 서버의 기능과 관련된 오류 ex) Port/IP 접속X

처리방안 : 연계 서버를 재시작 하거나 로그를 확인하여 원인을 분석한 후 처리

연계 데이터 오류 : 송신 시스템에서 생성 및 추출된 데이터의 값이 유효하지 않은 오류, **서버의 기능과 관련**

처리 방안 : 송신 연계 프로그램의 로그를 확인하여 데이터를 보정한 후 재전송

3. 장애 및 오류의 확인과 처리 절차

연계 서버에서 제공하는 장애 및 오류 현황 모니터링 화면을 통해 오류 원인 및 발생 현황 확인 → 연계 프로그램과 연계 서버에서 기록한 오류 로그 테이블 또는 파일을 확인하여 오류 원인을 분석 → 발생한 오류에 대한 원인 확인 시 원인에 따른 적절한 조치를 취함 → 필요한 경우 송신 시스템에서는 연계 데이터의 생성 또는 추출을 다시 수행하고 수신 시스템에서는 연계 데이터를 재 수신하여 운영 DB에 반영

4. 장애 및 오류의 정의와 설계

장애 및 오류 관리 대상 : 송 수신 시스템 오류 정의, 연계 서버는 연계 솔루션에 의해 관리되어 정의 필요x

관리 대상의 장애 및 오류 코드와 메시지 : 관리 대상에서 식별한 오류 내용을 주제별로 분류, 오류 내용에 **오류코드 부여, 오류 메시지를 정의**

장애 및 오류 코드와 메시지 관리 방식

테이블 관리 방식 : 관리 대상 오류 코드와 메시지가 많은 경우/ 파일 관리 방식 : 반대로 적은 경우

장애 및 오류 기록 방식 : 오류 로그 테이블 또는 파일은 기록 단위에 따라 인터페이스 테이블 또는 파일에 대한 로그, 연계 데이터에 대한 로그로 설계

44. 연계 데이터 보안 적용

1. 연계 데이터 보안의 개요 – 전송되는 연계 데이터는 보안에 취약해 데이터의 중요성을 고려해 보안을 적용

2. **전송 구간 보안** – 전송되는 데이터나 패킷을 쉽게 가로챌 수 없도록 암호화 기능이 포함된 프로토콜을 사용하거나 내용을 확인할 수 없게 암호화, VPN(가상 사설망), 연계 솔루션을 적용해 전송 구간 암호화 수행

3. **데이터 보안** – 송신 시스템에서 연계 데이터를 추출할 때와 수신 시스템에서 데이터를 운영 DB에 반영할 때 데이터를 암호화/복호화 하는 것

송신 시스템 : 추출된 연계 데이터를 암호화 알고리즘으로 암호화 후 인터페이스 테이블 또는 파일에 등록

수신 시스템 : 인터페이스 테이블 또는 파일의 데이터 중 암호화한 컬럼을 복호화 하여 운영 DB에 반영

4. 암 ∙ 복호화 적용 대상 선정 – 연계 데이터의 중요도에 따라 선정 ex) 개인정보

5. 암호화 알고리즘 – 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미

단방향 암호화 방식 : 해시 사용, 양방향 암호화 방식 : 개인키 및 공개키/알고리즘 : SEED, ARIA, DES, AES, RSA

6. 암 ∙ 복호화 적용

암호화 적용 대상, 암호화 알고리즘, 암호화 키(Key) 선정

암호화 적용 대상 컬럼(Column)의 데이터 길이 변경

암호화 알고리즘 라이브러리 확보 및 설치

연계 응용 프로그램에서 암 ∙ 복호화 처리 수행

45. 연계 모듈의 구현 환경

1. 연계 모듈 구현 환경의 개요 – 연계 메커니즘에 따라 구현된 연계 시스템 환경 및 모듈 구현 환경을 의미

2. EAI(Enterprise Application Intergration) – 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션

비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확장성(Determinacy)을 높여준다

구축 유형

Point – to – Point : 가장 기본적인 애플리케이션 통합 방식, 1:1 연결, 변경 및 재사용 어려움

Hub & Spoke : 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식, 확장 및 유지보수 용이

Message Bus(ESB방식) : 앱 사이에 미들웨어를 두어 처리하는 방식, 확장성이 뛰어나며 대용량 처리 가능

Hybrid : 허브와 버스의 혼합 방식, 그룹 내에서는 허브 방식을, 그룹 간에는 버스 방식을 사용

필요한 경우 한가지 방식으로 EAI 구현 가능, 데이터 병목 현상 최소화

3. ESB(Enterprise Service Bus) – 앱 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반 인터페이스 제공 솔루션

서비스 중심의 통합 지향, 범용적으로 사용하기 위해 결합도를 약하게 유지, 관리 및 보안 유지 수월, 품질↑

4. EAI, ESB 방식 환경 구축 절차

송 ∙ 수신 시스템에 연계를 위한 DB를 각각 설치하고 계정을 생성 → 데이터, 인터페이스, 매핑 정보, 오류 코드 등을 저장할 테이블을 생성 → 트리거 또는 프로그래밍 언어를 이용하여 연계 응용 프로그램을 구현

5. 웹 서비스(Web Service) – 네트워크의 정보를 표준화된 서비스 형태로 만들어 공유하는 기술, SOA 개념 실현

웹 서비스의 구성

SOAP(Simple Object Access Protocol) : HTTP, HTTPS, SMTP 등을 활용해 XML 기반의 메시지를 네트워크 상에서 교환하는 프로토콜

UDDI(Universal Description, Discovery and Integration) : WSDL을 등록하여 서비스, 서비스 제공자 검색 및 접근

WSDL(Web Service Description Language) : 웹 서비스명, 서비스 제공 위치, 프로토콜 등 **웹 서비스에 대한 상세 정보를 XML 형식으로 구현**

서비스 지향 아키텍처(SOA) : 정보 시스템을 공유와 재사용이 가능한 서비스 단위나 컴포넌트 중식으로 구축

6. 웹 서비스 방식 환경 구축 절차

1. 연계 파일, 로그파일, 오류 로그 파일 등이 생성될 위치와 파일명을 정의한 후 디렉터리 생성

2. 송신 시스템에서 DB로부터 데이터를 추출하여 XML 형식으로 변환하는 연계 응용 프로그램을 구현

3. WSDL의 내용을 바탕으로 전송된 파일 내용을 수신 시스템 DB에 반영하는 서비스를 실행

46. XML(eXtensible Markup Language)

1. XML의 개요 – 웹브라우저간 HTML 문법 호환x문제, SGML의 복잡함을 해결을 위해 개발된 다목적 마크업 언어

유니코드 기반으로 다국어 지원, 트리구조로 구성되 상위 태그는 여러 개의 하위 태그를 가질 수 있음, 텍스트 형식의 데이터 포맷으로 다양한 플랫폼과 시스템에서 사용 가능, 사용자가 직접 문서의 태그 정의 가능

마크업 언어 : 다른 문서의 처리를 위해 문서의 논리 구조나 체계를 정의하는 언어

2. XML의 구성 - <?xml version=”버전” encoding=”언어셋” standalone=”yes|no”?>

<?, ?> : 태그는 기본적으로 ‘<’,’>’를 사용하지만 문서의 첫 문단은 ? 기호를 포함한다.

Version : XML 문서에 사용된 XML의 버전을 명시

Encoding : XML의 언어셋(Character Set)을 명시, 생략 가능, 기본값 utf-8

Standalone : XML 문서를 해석할 때 외부 문서의 참조 여부 명시, 생략 가능, 기본값 no(외부 문서 참조)

XML 요소의 구성 - <요소이름 속성1=”속성값1” 속성2=”속성값2” …>내용 </요소이름>

요소 이름, 속성 = “속성값” : 요소와 관련된 속성과 속성값, 생략이 가능 / 내용 : 요소의 내용, 생략 가능

2. SOAP(Simple Object Access Protocol) – XML을 교환하기 위한 통신 규약

웹 서비스에서 사용되는 메시지의 형식과 처리 방법 지정, 기본적으로 HTTP 기반에서 동작하기 때문에 프록시와 방화벽의 영향 없이 통신 가능, XML과 동일한 텍스트 형식의 데이터 포맷으로 다양한 플랫폼과 시스템에서 활용 가능, 최근에는 무거운 구조의 SOAP 대신 RESTful 프로토콜을 이용하기도 함

RESTful 프로토콜 : HTTP와 REST(Representational State Transter)의 원칙을 사용하여 구현되는 웹 서비스, HTTP로 자원을 관리하는데 사용되는 웹 서비스 API의 집합을 의미

SOAP의 주요 요소

Envelope : XML 문서를 SOAP 메시지로 정의하는 것, 메시지에 대한 요소와 접근방법을 정의

Header : SOAP 메시지에 포함되는 웹 서비스를 정의하는 것, 생략이 가능

Body : 실제 SOAP 메시지가 포함된다

4. WSDL(Web Services Description Language)

웹 서비스와 관련된 서식이나 프로토콜 등을 표준적인 방법으로 기술하고 게시하기 위한 언어

XML로 작성, UDDI의 기초, SOAP/XML 스키마와 결합하여 인터넷에서 웹 서비스를 제공하기 위해 사용, 클라이언트는 XSDL 파일을 읽어 서버에서 어떠한 조작이 가능한지를 파악, 버전마다 요소들의 명칭이 다름

47. 연계 테스트

1. 연계 테스트의 개요 – 구축된 연계 시스템과 연계 시스템이 구성요소가 정상적으로 동작하는지 확인하는 활동

2. 연계 테스트 케이스 작성 – 시스템 간 데이터, 프로세스의 흐름을 분석해 필요 테스트 항목을 도출하는 과정

단위 테스트 케이스 : 확인해야할 항목 도출, 단순 개별 데이터의 유효 값을 확인/연관관계를 확인 구분 작성

연계 테스트 케이스 : 연계 응용 프로그램의 기능상 결함을 확인하는 단위 테스트 케이스 형태로 작성, 연계 테이블 간 송 수신 절차의 앞뒤로 연결하여 흐름을 확인할 수 있는 내용으로 작성

3. 연계 테스트 환경 구축 – 테스트의 일정, 방법, 절차, 소요 시간 등을 송 ∙ 수신 기관과의 협의를 통해 결정

연계서버/송 수신용 어댑터 설치, IP 및 포트 허용 신청, DB계정 및 테이블과 데이터 생성 등의 테스트 환경 구축

4. 연계 테스트 수행 – 연계 응용 프로그램을 실행해 테스트 케이스의 시험 항목 및 처리 절차 등을 실제로 진행

송 수신용 연계 응용 프로그램의 단위 테스트를 먼저 수행

단위 테스트의 수행 완료 후 연계 테스트 케이스에 따라 데이터 추출, 데이터 송 수신, 데이터 반영 과정 수행

5. 연계 테스트 수행 결과 검증

연계 테스트 케이스의 시험 항목 및 처리절차를 수행한 결과가 예상 결과와 동일한지를 확인하는 것이다

테스트 케이스 항목별 검증 방법

운영 DB 테이블의 건수 확인, 테이블 또는 파일을 열어 데이터를 확인, 파일 생성 위치에서 파일 생성 여부 및 파일 크기를 확인, 연계 서버에서 제공하는 모니터링 현황 확인, 시스템에서 기록하는 로그 확인