

통합 구현

통합 구현

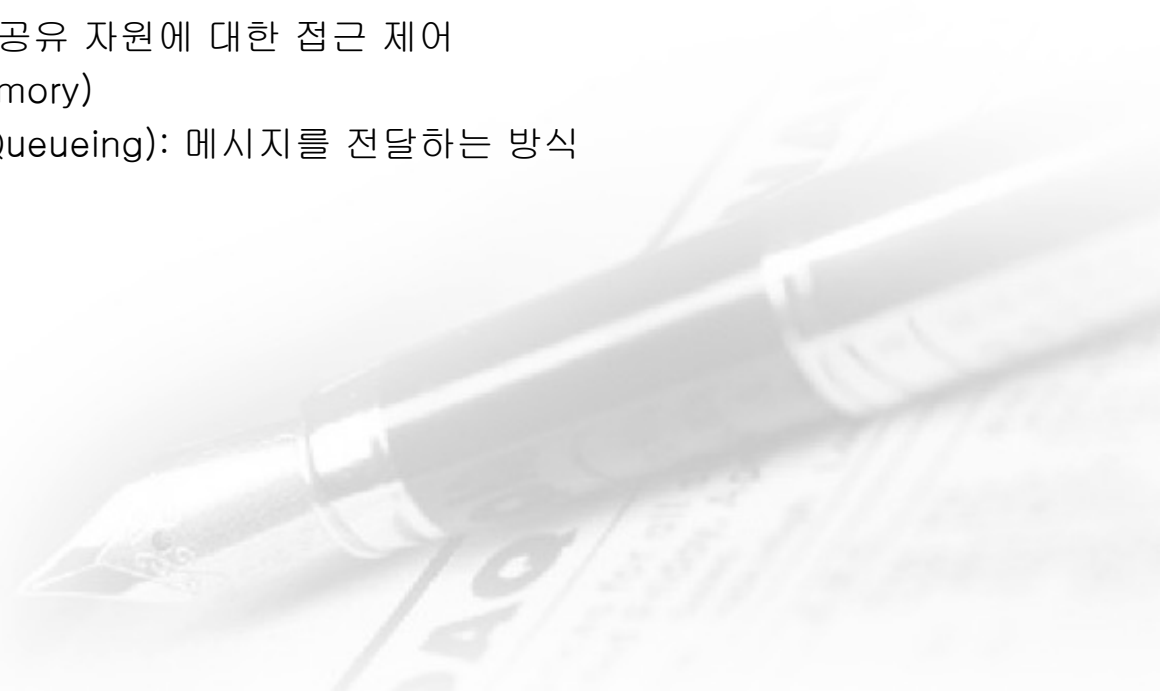
❖ 통합 구현

- ✓ 사용자의 요구사항에 맞춰 송·수신 모듈과 중계 모듈 간의 연계를 구현하는 것
- ✓ 사용자의 요구 사항을 해결하고, 새로운 서비스 창출을 위해 단위 기능을 하는 모듈 간의 연계와 통합
- ✓ 통합 구현은 송·수신 시스템과 모듈, 중계 시스템, 연계 데이터, 네트워크로 구성
 - 송신 시스템과 모듈
 - 송신 시스템은 데이터를 생성 및 변환하여 전송하는 시스템으로 송신 모듈과 모니터링 (Monitoring) 기능으로 구성
 - 송신 모듈은 전송 데이터를 생성하고 필요에 따라 전송 데이터의 변환 등을 수행
 - 모니터링 기능은 전송 데이터의 생성부터 송 신까지의 과정과 송신 상태 등을 확인
 - 수신 시스템과 모듈
 - 수신 시스템은 수신 받은 데이터를 정제 및 변환하는 시스템으로 수신 모듈과 모니터링 (Monitoring) 기능으로 구
 - 수신 모듈은 수신 데이터를 정제하고 애플리케이션이나 데이터베이스(DB) 테이블에 적합한 데이터로 변환하는 작업 등을 수행
 - 중계 시스템: 내·외부 시스템 간 또는 내부 시스템 간의 연계 시 사용되는 아키텍처
 - 연계 데이터: 송·수신 시스템 간 송·수신되는 데이터로 연계 데이터에는 속성, 길이(Size), 타입(Type) 등이 포함
 - 네트워크: 송신 시스템, 수신 시스템, 중계 시스템을 연결 해주는 통신망

IPC

❖ IPC(Inter-Process Communication)

- ✓ 모듈이나 프로세스 사이에 서로 데이터를 주고받는 행위 또는 그에 대한 방법이나 경로
- ✓ 구현 방식
 - 파일
 - 신호(Signal)
 - 소켓(Socket)
 - 메시지 큐
 - 파이프
 - 지명 파이프
 - 세마포어(Semaphore): 공유 자원에 대한 접근 제어
 - 공유 메모리(Shared Memory)
 - 메시지 전달(Message Queueing): 메시지를 전달하는 방식
 - (비공유)MPI 패러다임
 - 메모리 맵 파일



연계 요구사항 분석

❖ 연계 요구사항 분석

- ✓ 통합 구현을 위해 사용자 요구사항을 분석하고 연계 데이터를 식별 및 표준화하여 연계 데이터를 정의하는 것
- ✓ 연계 요구사항 분석 시 입력물
 - 시스템 구성도: 송수신 시스템의 네트워크, 하드웨어, 시스템 소프트웨어 구성
 - 응용 애플리케이션 구성: 송신 측에서 연계 데이터가 발생하는 응용 애플리케이션의 메뉴 구조도, 화면 및 화면 설계서를 이용해서 사용자 인터페이스 정의서, 연계할 데이터의 발생 시점 및 주기, 발생 패턴 등을 확인
 - 엔티티 관계도(ERD: Entity Relationship Diagram).테이블(공통 코드) 정의서 : 데이터 모델링 기술서, 테이블 간의 연관도, 테이블 별 칼럼(속성)이 정의된 테이블 정의서, 공통 코드 및 공통 코드 값에 대한 설명서를 이용해서 사용자 요구사항에서 데이터의 송수신 가능 여부, 데이터 형식 및 범위 등을 확인



연계 요구사항 분석

❖ 연계 요구사항 분석

- ✓ 요구사항 분석 시 도구 및 기법: 효과적으로 연계 요구 사항을 분석하기 위해 설문 조사, 인터뷰, 체크리스트(Checklist)와 같은 도구 및 기법을 활용
 - 사용자 인터뷰, 핵심 사용자 그룹 면담(FGI: Focus Group Interview)
 - 사용자 면담 또는 시스템 관리자 및 서비스 활용자와 같은 핵심 그룹 면담
 - 연계 데이터 정의, 연계 데이터의 활용 목적, 필요성 등을 식별하기 위함으로 사용자 인터뷰 전 연계 대상 시스템의 응용 애플리케이션 기능, 서비스의 확인이 필요함
 - (연계 분석) 체크리스트(Checklist): 연계 데이터와 연계 시스템 아키텍처 정의를 위해 시스템 운영 환경, 성능, 보안, 데이터 발생 등 다각도의 관점에서 고려 사항 점검 및 확인
 - 설문지 및 설문 조사
 - 서비스 활용 목적에 따라 연계가 필요한 데이터를 식별하고, 연계 주기 등을 분석하기 위해 설문 조사 항목을 통해 자료를 수집
 - 객관식 문항으로 예상 답변을 일정 범위 이내로 한정할 수도 있음
 - 델파이 기법: 통합 구현 및 연계 전문가, 시스템 아키텍처, 업무 전문가 등 각 분야 전문가로부터 연계 데이터 및 사용자 요구 사항 식별
 - 연계 솔루션(Solution) 비교 분석: EAI, ESB, Open API 등 다양한 연계 방식과 연계 솔루션 별 연계 시의 성능, 보안, 데이터 처리, 모니터링 등의 장단점을 비교함

연계 데이터 식별 및 표준화

❖ 연계 데이터 식별 및 표준화

- ✓ 연계 데이터를 구성하기 위해 연계 데이터를 식별하고 식별된 연계 데이터를 표준화한 후 이를 기반으로 연계(인터페이스) 정의서를 작성하는 과정
- ✓ 연계 데이터 식별 및 표준화 절차
 - 연계 범위 및 항목 정의: 데이터 타입, 길이, 코드화 여부 등 시스템 간 연계할 정보를 상세화 하여 정의
 - 연계 코드 변환 및 매핑: 정보로 전환하거나 검색 조건으로 활용하기 위해, 연계 정보 중 코드로 관리되어야 할 항목을 찾아 코드로 변환
 - 연계 데이터 와 변경 구분 추가: 수신 시스템 의 연계 정보에 송신 시스템의 식별자 (PK;Primary Key), 추가, 수정, 삭제 중 어떤 조작을 수행하는지 식별해주는 구분 정보를 추가하고, 연계 데이터의 확인 및 모니터링을 위해 인터페이스 테이블 또는 파일에 관리 정보를 추가함
 - 연계 데이터 표현 방법 정의: 연계 대상 범위 및 항목 과 코드 변환 및 매핑 방식을 정의한 이후에는 연계 데이터를 구성함
 - 연계 정의서 및 명세서 작성
 - 연계 정의서는 송·수신 시스템에서 연계할 항목 과 연계 데이터의 타입, 길이 등을 구성하고 형식을 정의한 결과를 기반으로 작성
 - 연계 정의서에는 송·수신 시스템의 시스템 명과 인터페이스 ID, 인터페이스 방식 및 주기 등 송·수신 시스템 간의 인터페이스 현황을 기재
 - 연계 명세서는 연계 정의서에 작성한 인터페이스 ID별로 인터페이스 주기 및 방식, DB 및 파일 형식, 송·수신 시스템의 타입, 길이, 식별자, 코드화 여부 등의 세부 사항을 작성

연계 메커니즘

❖ 연계 데이터 표준 항목

- ✓ 인터페이스 표준 항목은 송·수신 시스템을 연계하는데 표준적으로 필요한 데이터
- ✓ 시스템 공통부
 - 시스템 간 연동 시 필요한 공통 정보
 - 구성 정보에는 인터페이스 ID, 전송 시스템 정보, 서비스 코드 정보, 응답 결과 정보, 장애 정보 등
- ✓ 거래 공통부
 - 시스템들이 연동된 후 송·수신 되는 데이터를 처리할 때 필요한 정보
 - 구성 정보에는 직원 정보, 승인자 정보, 기기 정보, 매체 정보 등

❖ 연계 메커니즘

- ✓ 데이터의 생성 및 전송을 담당하는 송신 체계와 데이터 수신 및 운영 DB 반영을 담당하는 수신 체계로 구성
- ✓ 송신 시스템은 운영 DB로부터 인터페이스 테이블이나 파일(xml, text, csv 등) 형식으로 연계 데이터를 생성하여 송신
- ✓ 수신 시스템은 송신 시스템으로부터 전송된 데이터를 받아 수신 시스템에 맞는 데이터로 변환한 후 운영 DB에 반영
- ✓ 송·수신 시스템 사이에는 데이터의 송·수신과 송·수신 시스템 현황을 모니터링하는 중계 시스템을 설치
- ✓ 송·수신 시스템과 중계 시스템은 제각기 역할이 중복되지 않도록 아키텍처를 설계한 후 인터페이스 테스트와 통합 테스트를 통해 기능을 검증
- ✓ 연계 메커니즘의 연계 방식에는 직접 연계 방식과 간접 연계 방식이 있다.

연계 메커니즘

❖ 연계 메커니즘

✓ 직접 연계 방식

- 중간 매개체 없이 송·수신 시스템이 직접 연계하는 방식
- 장점
 - 연계 및 통합 구현이 단순
 - 중간 매개체 없이 직접 연계되기 때문에 데이터 연계처리 성능이 우수
 - 개발 비용이 저렴
 - 개발 기간이 짧음
- 단점
 - 시스템 변경 시 오류가 발생할 수 있음
 - 암호화와 복호화가 어려움
 - 비즈니스 로직을 적용하기 어려움
 - 연계 및 통합이 가능한 시스템 환경이 제한적
- 구현 방식
 - DB Link: 데이터베이스의 링크를 제공
 - DB Connection Pool(WAS): WAS가 데이터베이스 연결을 제공
 - JDBC: Java에서 데이터베이스에 접근하기 위한 API
 - 화면 링크: Web Application 화면에서 하이퍼링크 이용
 - Open API: 데이터를 읽어서 제공하는 인터페이스

연계 메커니즘

❖ 연계 메커니즘

✓ 간접 연계 방식

- 송·수신 시스템 사이에 중간 매개체를 두어 연계하는 방식
- 장점
 - 서로 다른 네트워크, 프로토콜 등 다양한 환경의 연계 및 통합이 가능
 - 송수신 시스템 간 인터페이스가 변경되어도 오류 없이 서비스 가능
 - 보안 품질 보장과 비즈니스 로직을 쉽게 반영
- 단점
 - 연계 메커니즘이 복잡
 - 중간 매개체로 인한 성능 저하
 - 개발 및 적용 기간이 길어짐



연계 메커니즘

❖ 연계 메커니즘

✓ 간접 연계 방식

○ 구현 방식

▪ 연계 솔루션

❑ EAI 서버와 송·수신 시스템에 설치되는 클라이언트(Client)를 이용하는 방식

❑ EAI: 송·수신 데이터를 식별하기 위해 송·수신 처리 및 진행 현황을 모니터링하고 통제하는 시스템

▪ ESB(Enterprise Service Bus): 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 방식

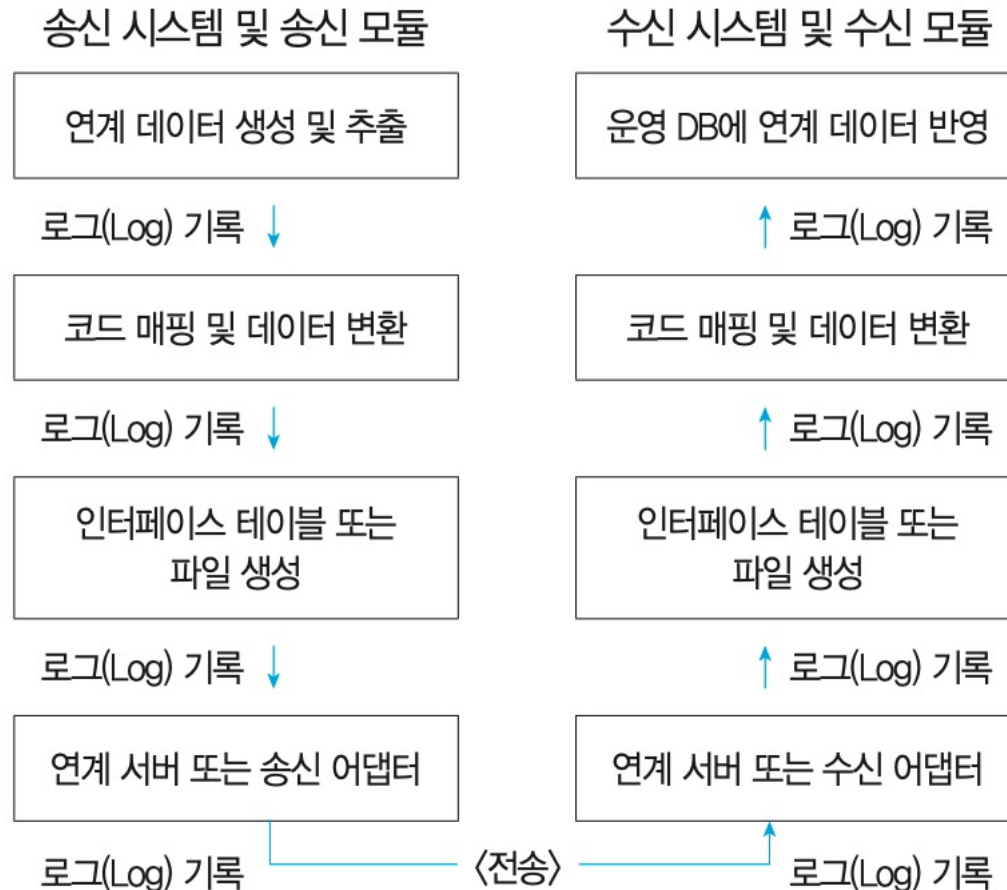
▪ Socket: 서버는 통신을 위한 소켓(Socket)을 생성하여 포트를 할당하고 클라이언트의 통신 요청 시 클라이언트와 연결하여 통신하는 네트워크 기술

▪ Web Service: 웹 서비스(Web Service)에서 WSDL과 UDDI, SOAP 프로토콜을 이용하여 연계하는 방식



연계 메커니즘 구성

❖ 연계 메커니즘 구성



연계 메커니즘 구성

❖ 연계 메커니즘 구성

- ✓ 연계 데이터 생성 및 추출: 연계 솔루션과 관계없이 응용 프로그램이나 DB 등 응용 시스템에서 연계 데이터를 생성하고 추출하는 것
- ✓ 코드 매핑 및 데이터 변환: 송신 시스템에서 사용하는 코드를 수신 시스템에서 사용하는 코드로 매핑 및 변환하는 것
- ✓ 인터페이스 테이블 또는 파일 생성: 연계 데이터를 인터페이스 테이블이나 파일 형식으로 생성하는 것
- ✓ 로그 기록: 송·수신 시스템에서 수행되는 모든 과정에 대한 결과 및 오류에 대한 정보를 로그 테이블이나 파일에 기록하는 것
- ✓ 연계 서버 또는 송·수신 어댑터
 - 연계 서버: 송·수신 시스템 중 한 곳에 설치하며 인터페이스 테이블 또는 파일의 데이터를 전송 형식에 맞게 변환하고 송·수신을 수행하는 등 송수신과 관련된 모든 처리 수행
 - 송신 어댑터 : 송신 시스템에 설치하며, 인터페이스 테이블 또는 파일의 데이터를 전송 형식에 맞도록 변환하고 송신 수행
 - 수신 어댑터 : 수신 시스템에 설치하며, 송신 시스템으로부터 수신한 데이터를 인터페이스 테이블이나 파일로 생성
- ✓ 전송: 송신 시스템에서 생성된 연계 데이터를 네트워크 환경에 맞는 데이터로 변환한 후 수신 시스템으로 보내는 것
- ✓ 운영 DB에 연계 데이터 반영: 수신된 인터페이스 테이블 또는 파일 구조의 데이터를 변환 프로그램을 이용하여 수신 시스템의 운영 DB에 반영하는 것

연계 장애 및 오류처리 구현

❖ 연계 메커니즘 구간별 장애 및 오류 모니터링 현황

구간	오류 발생 시점	오류 로그 기록 장소	오류 로그 기록 주체
송신 시스템	<ul style="list-style-type: none"> • 데이터 생성 및 추출 시 • 코드 매핑 및 데이터 변환 시 • 인터페이스 테이블 또는 파일 등록 시 	송신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블	별도로 구현한 송신용 연계 프로그램
수신 시스템	<ul style="list-style-type: none"> • 연계 데이터 로드 (Load) 시 • 코드 매핑 및 데이터 변환 시 • 운영 DB에 반영 시 	수신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블	별도로 구현한 수신용 연계 프로그램
연계 서버	<ul style="list-style-type: none"> • 연계 데이터 로드 (Load) 및 전송 형식으로 변환 시 • 연계 데이터 송·수신 시 • 수신 시스템의 데이터 형식으로 변환 및 로드 (Load) 시 	연계 서버에 설정된 로그 및 오류 로그 테이블 또는 파일	연계 서버

연계 장애 및 오류처리 구현

❖ 장애 및 오류 유형과 처리 방안

오류 유형	내용	처리방안
송신 시스템의 연계 프로그램	데이터 생성 및 추출 권 한이 없거나 데이터 변 환 시 예외 (Exception) 미처리 등 연계 프 로그램 구현 상의 오류	송·수신 연계 프로그램의 로그 를 확인하여 원인을 분석한 후 처리 하고, 데이터 전송이나 반 영을 재작업
수신 시스템의 연계 프로그램	운영 DB 접근 권한이 없거나 데이터 변환 및 반영 시 예외 (Exception) 미처리 등 연계 프 로그램 구현 상의 오류	
연계 서버	연계 서버 실행, 송·수 신, 전송 형식 변환 등 서버의 기능과 관련된 오류	<ul style="list-style-type: none"> •연계 서버를 재시작 •연계 서버의 로그를 확인하여 원인을 분석한 후 처리
연계 데이터	송신 시스템에서 생성 및 추출 된 데이터의 값이 유효하지 않 은 오류	송신 연계 프로그램의 로그를 확인하여 데이터를 보정한 후 재전송

연계 장애 및 오류처리 구현

❖ 장애 및 오류의 확인과 처리 절차 / 정의와 설계

✓ 장애 및 오류의 확인과 처리절차

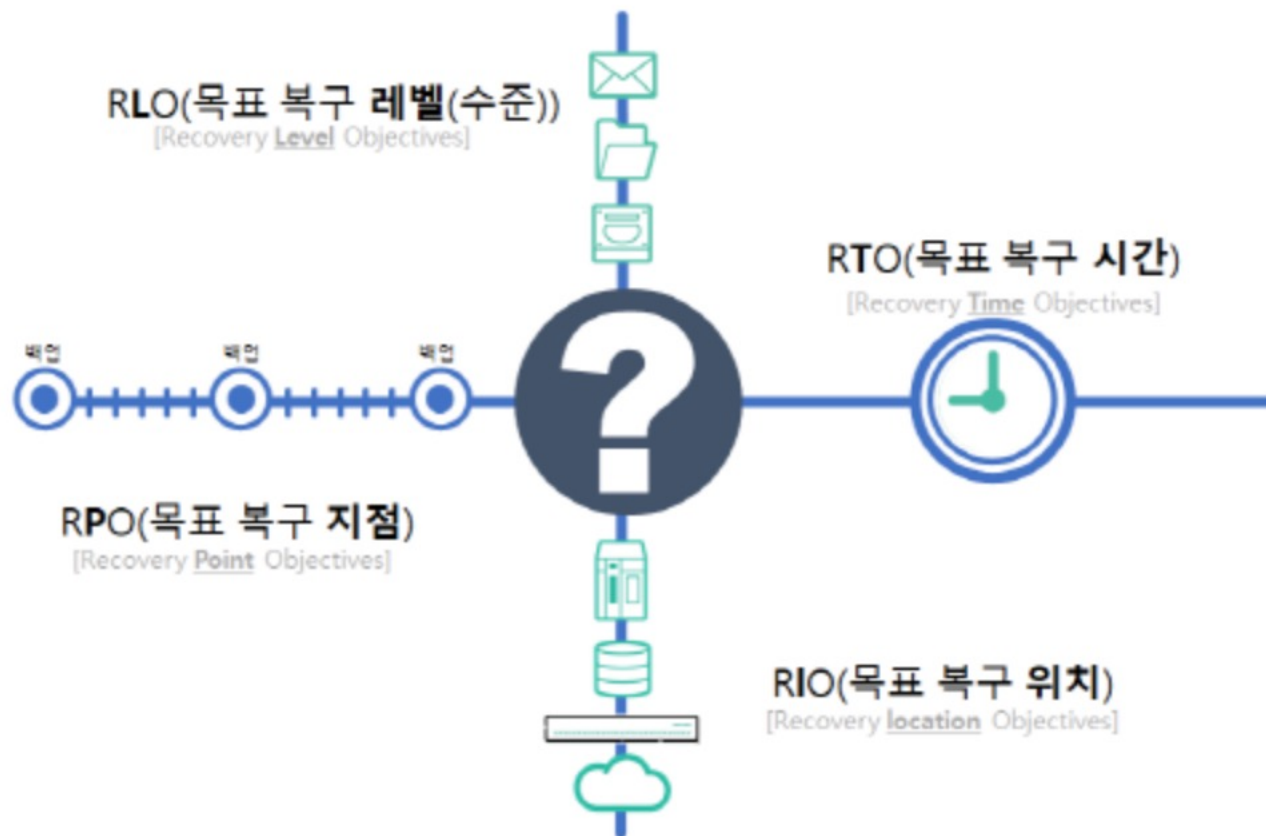
- 장애 및 오류는 1차적으로 연계 서버에서 제공하는 장애 및 오류 현황 모니터링 화면을 통해 오류 원인 및 발생 현황을 확인
- 1차에서 확인이 불가능한 경우 송·수신 시스템의 연계 프로그램과 연계 서버에서 기록한 오류 로그 테이블 또는 파일을 확인하여 오류 원인을 분석
- 발생한 오류에 대한 원인이 확인되면 원인에 따른 적절한 조치를 취한다.

✓ 장애 및 오류의 정의와 설계

- 장애 및 오류 관리 대상: 송·수신 시스템의 연계 프로그램에서 관리하는 장애 및 오류를 관리 대상으로 정의
- 관리 대상의 장애 및 오류 코드와 메시지: 관리 대상에서 식별한 오류 내용을 주제별로 분류한 후 각 오류 내용에 오류 코드를 부여하고 오류 메시지를 정의
- 장애 및 오류 코드와 메시지 관리 방식
 - 오류 코드와 오류 메시지를 관리하는 방식에는 테이블 관리 방식과 파일 관리 방식이 있음
 - 테이블 관리 방식은 관리 대상 오류 코드와 오류 메시지가 많은 경우 파일 관리 방식은 관리 대상 오류 코드와 오류 메시지가 적은 경우 사용
- 장애 및 오류 기록 방식: 오류 로그 테이블 또는 파일은 기록 단위에 따라 인터페이스 테이블 또는 파일에 대한 로그, 연계 데이터에 대한 로그로 설계

연계 장애 및 오류처리 구현

❖ RTO, RPO, RIO, RLO



연계 장애 및 오류처리 구현

❖ RTO, RPO, RIO, RLO

- ✓ RPO (Recovery Point Objective): 목표 복구 시점
 - 장애 발생 시 비즈니스 연속을 위해 어느 시점으로 백업할지 결정하게 될 지표
 - 오늘날 IT기술에서 일 단위의 RPO를 사용하는 것은 무리가 있고 분 단위, 초 단위의 RPO 지표를 고려
 - 데이터 손실을 얼마나 감당할 수 있는가?
 - 감내할 수 있는 데이터 손실의 양(시간)
 - 현재로부터 가장 가까운 복원지점(백업시점) 까지의 시간 목표
- ✓ RTO (Recovery Time Objective): 목표 복구 시간
 - 장애 발생 시 시스템을 원상태로 복원하는데 소요되는 시간
 - 시스템이 얼마나 빨리 복구되어야 하는가?
 - 서비스가 재개될 때까지 걸리는 시간(목표)
- ✓ RIO (Recovery location Objection): 목표 복구 위치
 - 전통적으로 데이터 백업은 테이프에 기록했다.
 - 최신의 트렌드는 디스크에 백업하거나 클라우드를 활용하여 백업하는 것이다.
- ✓ RLO(Recovery Level Objective):목표 복구 레벨(수준)
 - 정보 시스템은 OS, 파일 데이터, 데이터베이스, 애플리케이션으로 나눌 수 있는데 대부분의 정보 시스템은 파일 데이터와 데이터베이스 만을 복원했다 그러나 최신의 기술로는 대부분 복원 가능하다.

연계 데이터 보안

❖ 연계 데이터 보안

- ✓ 송신 시스템에서 수신 시스템으로 전송되는 연계 데이터는 보안에 취약할 수 있으므로 데이터의 중요성을 고려하여 보안을 적용
- ✓ 일반적으로 연계 데이터의 보안은 전송 구간에서의 암호·복호화 와 데이터의 암호·복호화로 구현
- ✓ 전송 구간 보안
 - 전송되는 데이터나 패킷(Packet)을 쉽게 가로챌 수 없도록 암호화 기능이 포함된 프로토콜을 사용하거나 데이터나 패킷을 가로채더라도 내용을 확인할 수 없게 데이터나 패킷을 암호화
 - 전송 구간 암호화를 지원하는 VPN(가상 사설망) 이나 연계 솔루션을 적용해 전송 구간 암호화를 수행
- ✓ 데이터 보안
 - 송신 시스템에서 연계 데이터를 추출할 때와 수신 시스템에서 데이터를 운영 DB에 반영할 때 데이터를 암호·복호화 하는 것
 - 데이터 보안을 위해서는 암호·복호화 적용 대상 선정, 암호화 알고리즘, 암호·복호화 적용을 위한 환경설정을 설계 및 구현
 - 암호·복호화 적용 대상 선정 : 연계 데이터의 중요도에 따라 선정
 - 송·수신 시스템에 정의된 기준에 따라 다르지만 일반적인 적용 대상은 주민등록번호, 운전면허번호, 장애인관리번호, 은행계좌번호, 신용카드번호 등 ‘개인 정보 보호법’에 근거한 개인 정보

연계 데이터 보안

❖ 연계 데이터 보안

✓ 데이터 보안

○ 암호화 알고리즘 : 기준에 따라 다양하게 분류

- 단방향(암호화는 가능하나 복호화는 불가능) 과 양방향 알고리즘
- 대칭 키 알고리즘 과 비대칭 키 알고리즘으로 구분할 수 있는데 대표적인 대칭 키 알고리즘으로 국내에서 개발한 방법으로는 SEED-128, SEED-256, ARIA 가 있으며 국외에서는 AES128, AES192, AES256, DES, 3DES 등이 있음



연계 모듈의 구현 환경

❖ 연계 모듈의 구현 환경

- ✓ 연계 모듈의 구현 환경 • 연계 메커니즘에 따라 구현된 연계 시스템 환경 및 모듈 구현 환경
- ✓ 연계 모듈의 구현 환경은 트리거(Trigger) 또는 프로그래밍 언어를 이용한 EAI, ESB 방식과, 배치 프로그램 및 인터페이스 파일을 이용한 웹 서비스(Web Service) 방식으로 구분
- ✓ EAI (Enterprise Application Integration)
 - 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해 주는 솔루션
 - EAI의 구축 유형
 - Point-to-Point: 가장 기본적인 애플리케이션 통합 방식으로 애플리케이션을 1 : 1로 연결
 - Hub & Spoke: 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형
 - Message Bus(ESB 방식): 애플리케이션 사이에 미들웨어를 두어 처리하는 방식
 - Bus Hybrid : Hub & Spoke와 Message Bus 의 혼합 방식
- ESB (Enterprise Service Bus)
 - 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션
 - ESB는 애플리케이션 통합 측면에서 EAI와 유사하지만 애플리케이션 보다는 서비스 중심의 통합을 지향

연계 모듈의 구현 환경

❖ 연계 모듈의 구현 환경

✓ 웹 서비스 (Web Service)

- 네트워크의 정보를 표준화된 서비스 형태로 만들어 공유하는 기술로 서비스 지향 아키텍처(SOA) 개념을 실현하는 대표적인 기술
- 웹 서비스의 구성
 - SOAP(Simple Object Access Protocol): HTTP, HTTPS, SMTP 등을 활용하여 XML 기반의 메시지를 네트워크 상에서 교환하는 프로토콜
 - UDDI(Universal Description, Discovery and Integration): WSDL을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용됨
 - WSDL(Web Services Description Language): 웹 서비스 명, 서비스 제공 위치, 프로토콜 등 웹 서비스에 대한 상세 정보를 XML 형식으로 구현



데이터 포맷

❖ XML(eXtensible Markup Language)

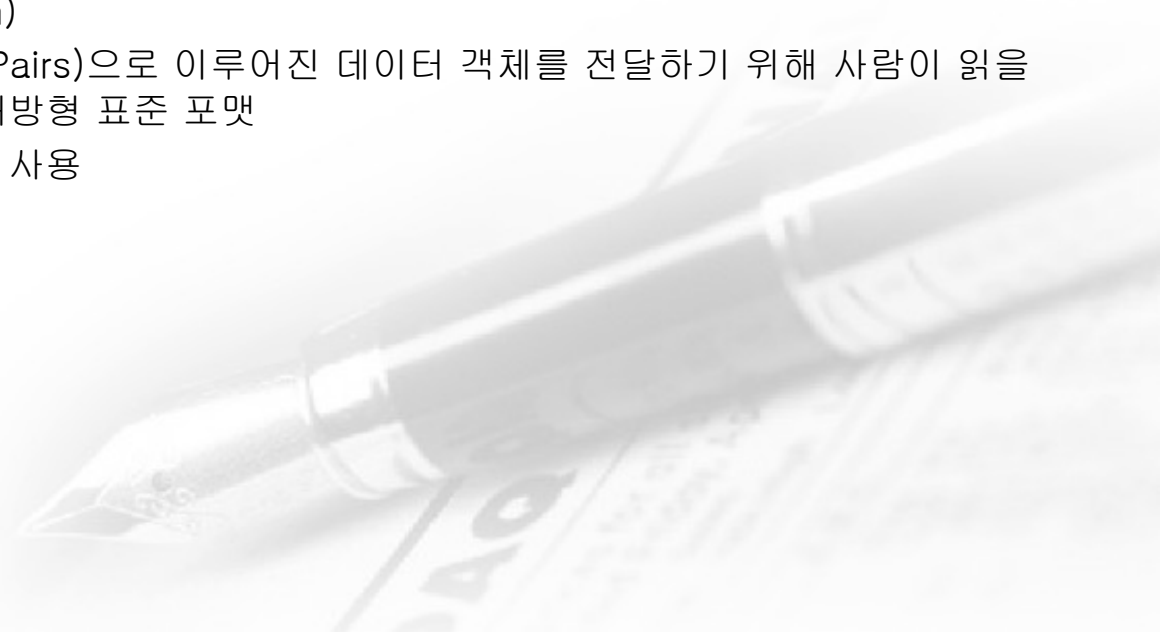
- ✓ 웹 브라우저 간 HTML 문법이 호환되지 않는 문제와 SGML의 복잡함을 해결하기 위하여 개발된 다목적 마크업 언어
- ✓ 유니코드를 기반으로 다국어를 지원
- ✓ 사용자가 직접 문서의 태그를 정의할 수 있으며, 다른 사용자가 정의한 태그를 사용할 수 있다.

❖ CSV(comma-separated values)

- ✓ 구분자로 구분한 텍스트

❖ JSON(JavaScript Object Notation)

- ✓ 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷
- ✓ XML을 대체하여 최근에 많이 사용



통신 규약

- ❖ SOAP(Simple Object Access Protocol): 일반적으로 널리 알려진 HTTP, HTTPS, SMTP 등을 통해 XML 기반의 메시지를 컴퓨터 네트워크 상에서 교환하는 프로토콜로 웹 서비스에서 기본적인 메시지를 전달하는 기반
- ❖ WSDL(Web Service Description Language): 웹 서비스 기술 언어 또는 기술된 정의 파일의 총칭으로 XML로 기술되며 웹 서비스의 구체적 내용이 기술되어 있어 서비스 제공 장소, 서비스 메시지 포맷, 프로토콜 등을 기술
- ❖ RESTful: 웹 상의 자료를 HTTP위에서 SOAP이나 쿠키를 통한 세션 트래킹 같은 별도의 전송 계층 없이 전송하기 위한 아주 간단한 인터페이스
- ❖ 링크드 데이터(linked data): 웹 상에 존재하는 데이터를 개별 URI(Uniform Resource Identifier)로 식별하고, 각 URI에 링크 정보를 부여함으로써 상호 연결된 웹을 지향하는 모형이다. 링크 기능이 강조된 시맨틱웹의 모형에 속한다고 볼 수 있으며 팀 버너스 리의 W3C를 중심으로 발전하고 있다.
 - ✓ URI의 사용: 링크드 데이터는 웹문서의 위치를 나타내는 URL 중심의 식별 체계를 지양하는 것으로 개별 문서에 존재하는 개별 객체에 각각 URI를 부여하는 것
 - ✓ HTTP URI의 사용: 링크드 데이터는 URI 중에서도 HTTP 프로토콜을 통해 접근할 수 있는 URI를 제안하고 있는데 이는 링크드 데이터에 대한 접근성을 강화하려는 목적
 - ✓ RDF의 사용: 링크드 데이터는 RDF와 같이 트리플 모형으로 구조화된 데이터를 사용
 - ✓ 링크 정보의 부여: 위의 세 가지 원칙을 지켰다고 해도, 풍부한 링크 정보가 없다면 웹에 존재하는 데이터를 연결하기는 어렵기 때문에 기존의 시맨틱 웹에서도 링크 정보를 부여할 수는 있었으나, 그 보다는 구조화된 표현을 주로 강조했으며 링크드 데이터에서는 보다 발전된 시맨틱 웹을 위해 링크 정보를 부여하는 것이 매우 중요

Middle Ware

❖ 미들웨어

- ✓ 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 추가적인 서비스를 제공하는 소프트웨어
- ✓ 종류
 - DB(DataBase): 데이터베이스 벤더에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어
 - RPC(Remote Procedure Call): 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어
 - MOM(Message Oriented Middleware)
 - 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어
 - 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용
 - TP-Monitor(Transaction Processing Monitor)
 - 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어
 - 사용자 수가 증가해도 빠른 응답 속도를 유지해야 하는 업무에 주로 사용
 - ORB(Object Request Broker)
 - 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어
 - TP-Monitor의 장점인 트랜잭션 처리와 모니터링 등을 추가로 구현한 제품도 있다.
 - WAS(Web Application Server): 정적인 콘텐츠를 처리하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어

연계 테스트

❖ 연계 테스트

- ✓ 구축된 연계 시스템과 연계 시스템의 구성 요소가 정상적으로 동작하는지 확인하는 활동
- ✓ 연계 테스트 절차
 - 연계 테스트 케이스 작성: 연계 시스템 간의 데이터 및 프로세스의 흐름을 분석하여 필요한 테스트 항목을 도출하는 과정
 - 연계 테스트 환경 구축: 테스트의 일정, 방법, 절차, 소요 시간 등을 송·수신 기관과의 협의를 통해 결정
 - 연계 테스트 수행: 연계 응용 프로그램을 실행하여 연계 테스트 케이스의 시험 항목 및 처리 절차 등을 실제로 진행
 - 연계 테스트 수행 결과 검증: 연계 테스트 케이스의 시험 항목 및 처리 절차를 수행한 결과가 예상 결과와 동일한지를 확인

