
통합구현

2001020206_16v4

- 통합구현은 사용자의 요구사항을 해결하고 새로운 서비스 창출을 위해 단위 기능을 하는 모듈 간의 연계와 통합이다.
- 통합구현은 시스템 구성, 데이터 송수신 방식, 송수신 모듈 구현 방법에 따라 다양하므로, 구축하고자 하는 환경과 사용자 요구사항에 따라 적합한 통합 구현 방법을 설계한다.



1

연계 데이터 구성하기

2

연계 매커니즘 구성하기

3

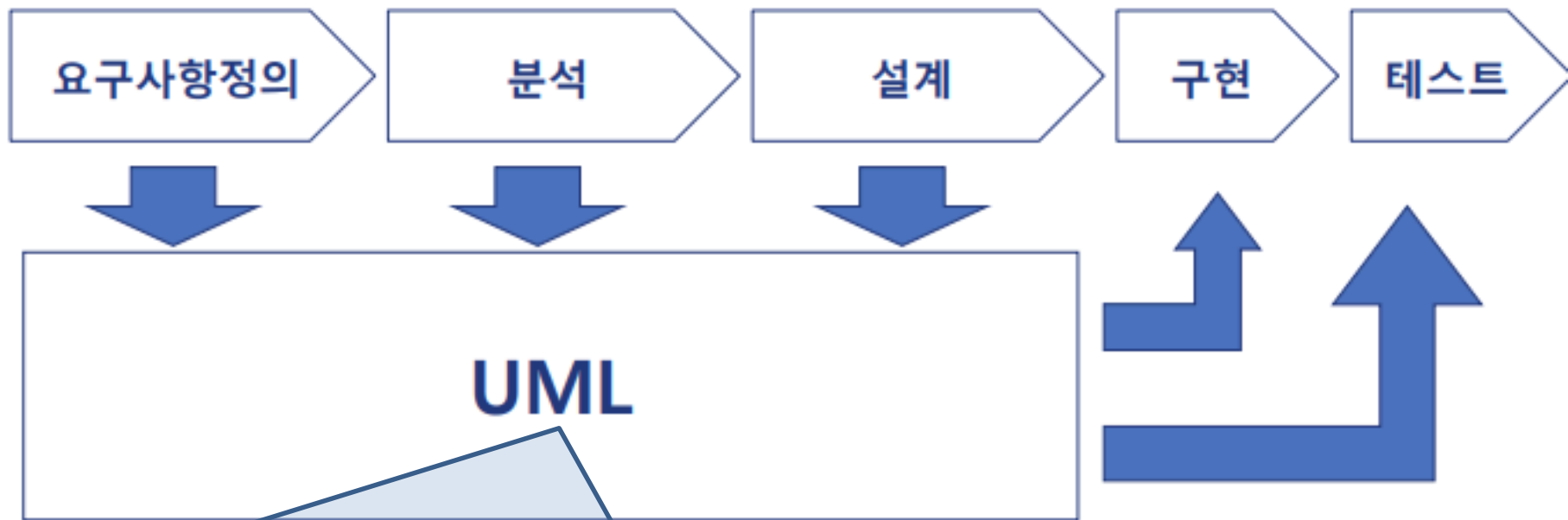
내외부 연계 모듈 구현하기

1

연계 데이터 구성하기

UML

❖ 개발의 전 과정에서 사용됨



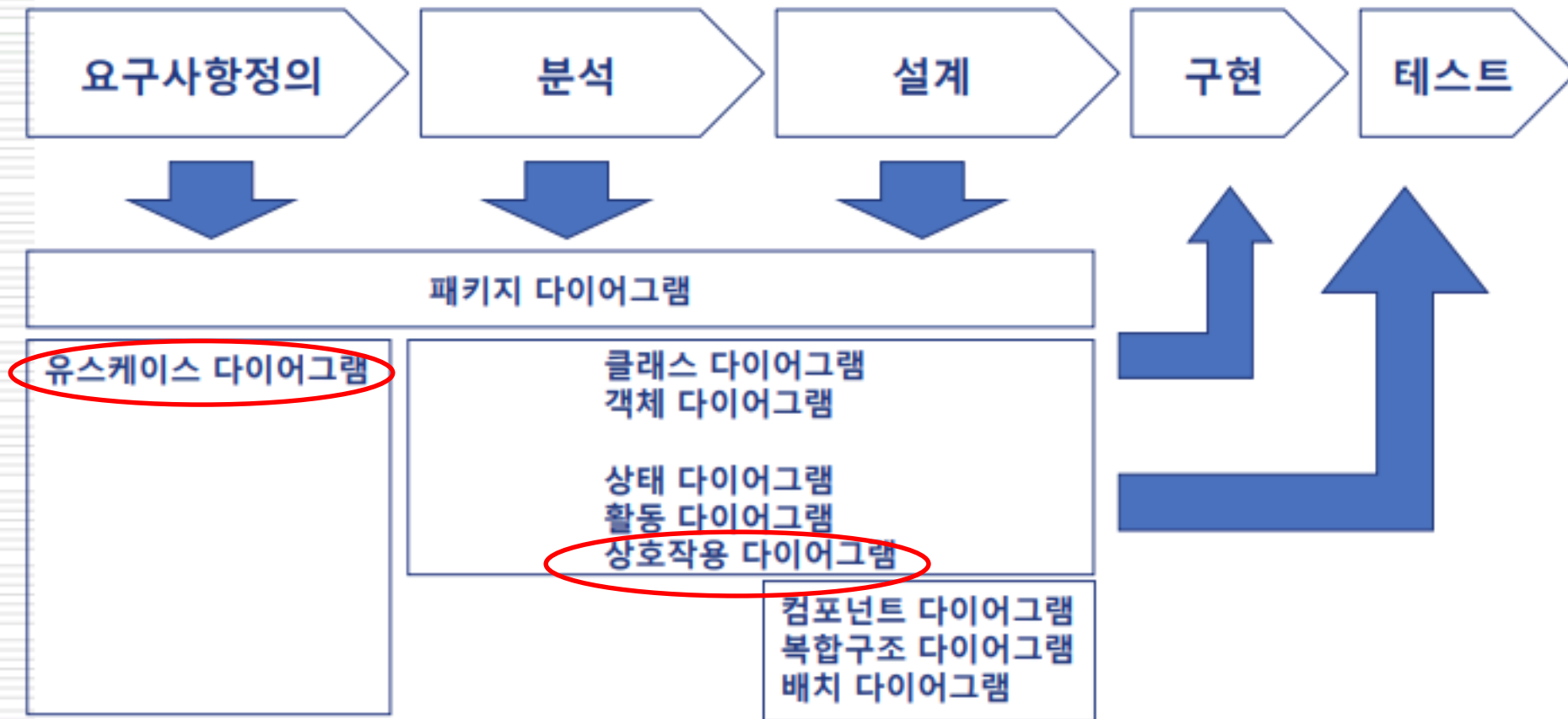
시스템 개발자가 구축하고자 하는 소프트웨어를 코딩하기에 앞서 표준화되고 이해되기 쉬운 방법으로 소프트웨어를 설계하여 다른 사람들과 효율적으로 의사소통할 수 있는 메커니즘을 제공한다.

ERD

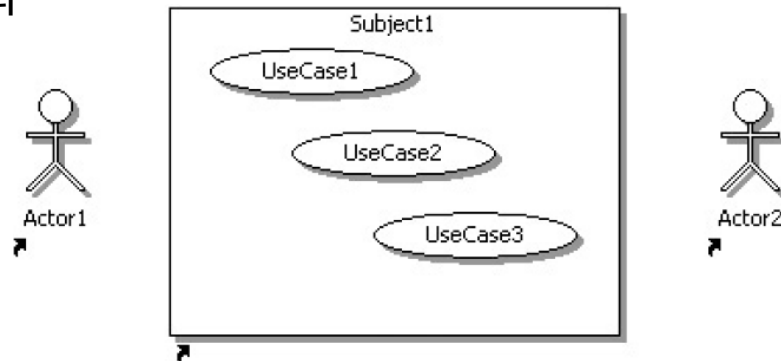
데이터 모델링 기술서, 테이블 간 연관도, 컬럼(속성)이 정의된 테이블 정의서, 테이블 형식 및 범위등을 확인할 수 있는 시스템 요구사항 분석서의 입력물

UML

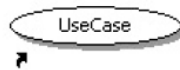
❖ 개발 단계 별로 주로 사용되는 다이어그램이 있음



- 시스템 개발에 참여하는 사람들을 큰 부류로 나누면,
 - ✓ 의뢰인, 개발팀, 사용자 (User)
- 사용자의 관점을 빨리 이해해야만 쓸모 있고 유용한 시스템을 만들 수 있다.
- 개발자는 가능한 한 모든 요구사항을 파악하여 사용자의 승인을 받아야만 후일 요구사항 변경에 대한 위험부담을 줄일 수 있다.
- [그림 3-1]
 - ✓ 네모난 창은 시스템의 경계
 - ✓ UseCase1, 2, 3은 구축할 시스템의 기능을 표현
 - ✓ 유스케이스 모델링에서 개발할 시스템 외부 존재를 Actor라는 개념으로 정의
 - ✓ 시스템의 범위에 해당되어 개발될 시스템의 단위 기능을 Usecase 개념으로 정의

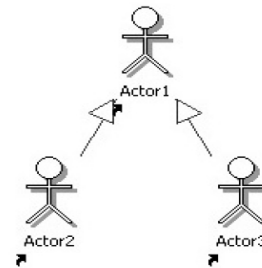
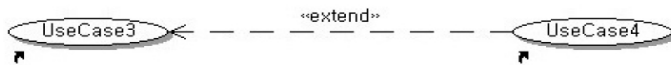
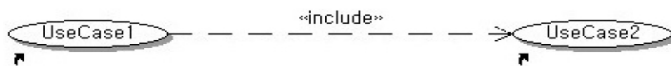


- 유스케이스의 역할 : 사용자의 시점에서 시스템을 모델링한다는 것
- 유스케이스 : 시스템에 대한 시나리오의 집합
- 시나리오 : 발생하는 이벤트의 흐름이 나타나 있다.
- 이벤트의 흐름 : 사람, 시스템, 하드웨어, 혹은 시간의 흐름에 의해 시작
- 액터 : 이벤트 흐름을 시작하게 하는 객체



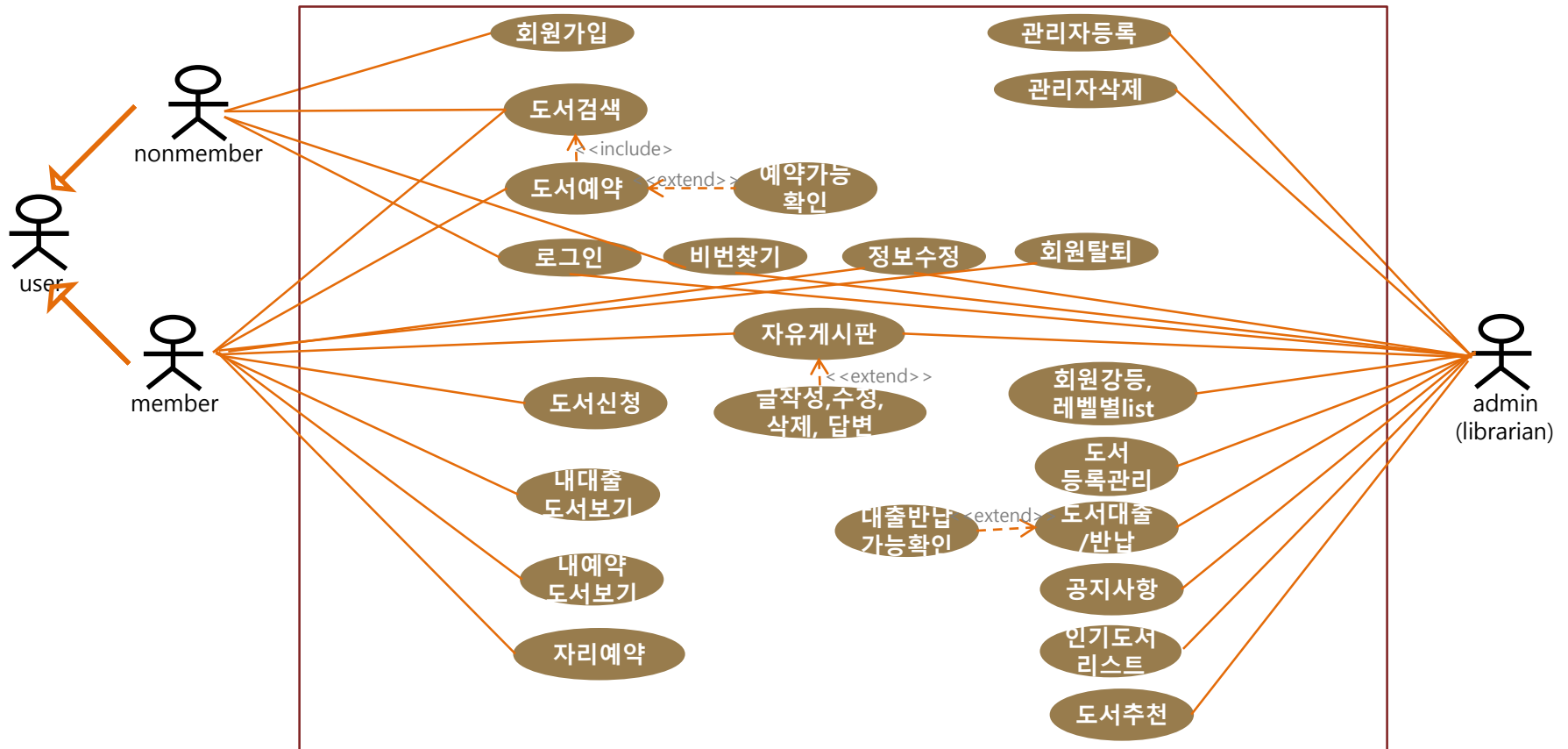
[유스케이스와 액터]

- 유스케이스 사이에는 일반적인 연관관계 이외에 또 다른 관계
 - ✓ 포함(Include)관계 : 다른 유스케이스에서 기존의 유스케이스를 재사용하는 관계
 - ✓ 확장(Extend)관계 : 기존의 유스케이스에 진행단계를 추가하여 새로운 유스케이스를 만들어내는 관계
 - ✓ 액터와 유스케이스에 대한 일반화관계



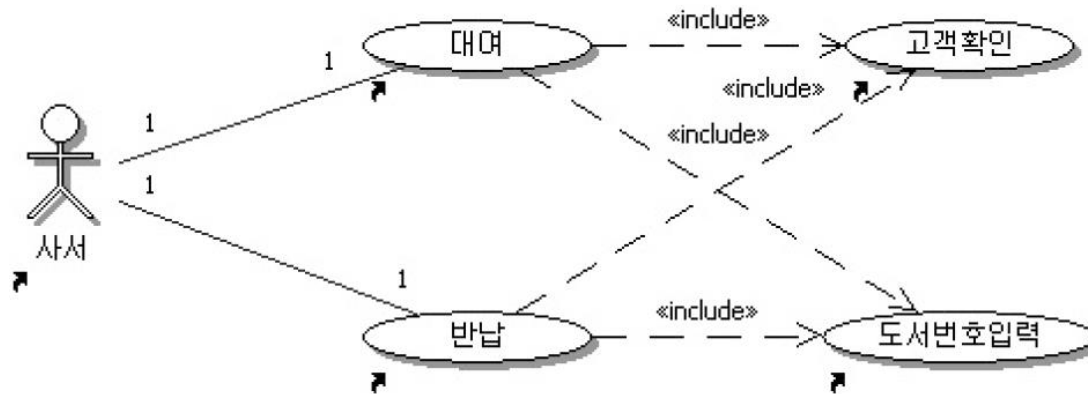
[유스케이스 관계]

도서관 시스템의 유스케이스 다이어그램 예제



● 유스케이스간 포함관계

- 포함관계 : 유스케이스를 수행할 때 다른 유스케이스가 반드시 수행되는 것
- 유스케이스 다이어그램에서는 다른 유스케이스가 나타내는 이벤트 흐름을 포함(include)하는 관계를 유스케이스간에 표현



[도서관 시스템의 포함관계]

[표] 포함관계를 이용한 '대여'와 '반납' 유스케이스의 이벤트 흐름

대여 유스케이스 이벤트 흐름	반납 유스케이스 이벤트 흐름
<ol style="list-style-type: none"> 1. '고객확인' 유스케이스를 포함한다. 2. 사서는 '대여'를 선택한다. 3. '도서번호입력' 유스케이스를 포함한다. 4. 도서관 시스템은 고객이 대여가 가능한지 확인한다. 5. 고객에게 대여를 가능 여부를 표시한다. 6. 도서관 시스템은 도서를 대여 처리한다. 	<ol style="list-style-type: none"> 1. '고객확인' 유스케이스를 포함한다. 2. 사서는 '반납'을 선택한다. 3. '도서번호입력' 유스케이스를 포함한다. 4. 도서관 시스템은 도서를 반납 처리한다.

● 유스케이스간 확장관계

- 확장관계의 유스케이스는 포함관계처럼 여러 유스케이스에 걸쳐 중복적으로 사용되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것을 의미
- 확장(extend)하는 유스케이스는 상위 유스케이스로부터 어떠한 특정 조건에 의해 수행됨을 의미
- 포함관계와 확장관계의 차이점
 - ✓ 포함관계 : 여러 유스케이스에서 공통적으로 발견되는 기능을 표현
 - ✓ 확장관계 : 한 유스케이스에서 추가되거나 확장된 기능을 표현



[금액계산 유스케이스의 확장]

[표] 금액계산 유스케이스와 금액표시 유스케이스의 이벤트 흐름

금액계산 유스케이스 이벤트 흐름	금액표시 유스케이스 이벤트 흐름
1. 사용자가 자판기에 동전을 투입하거나 음료수를 선택한다. 2. 현재 금액에 투입된 동전만큼 액수를 추가한다. 3. 금액표시 유스케이스 확장	1. 금액계산 유스케이스로부터 금액을 받는다. 2. 받은 금액을 표시한다.

- 순차 다이어그램(Sequence Diagram)
 - ✓ 객체간의 동적 상호작용을 시간적 개념을 중심으로 모델링하는 과정
 - ✓ 다이어그램의 수직방향이 시간의 흐름을 나타낸다.
- 순차 다이어그램은 유스케이스를 실현(realization)한다.
 - ✓ 유스케이스 다이어그램에서는 시스템이 제공해야 하는 서비스를 정의하기 때문에 유스케이스는 프로그램으로 구현되기 전에 순차 다이어그램으로 설계
- 순차 다이어그램(Sequence Diagram)은 상호작용 다이어그램(Interaction Diagram)으로 불리며 시스템의 동적 측면을 모델링하기 위해 UML에서 사용하는 다이어그램
 - ✓ 순차 다이어그램은 메시지의 시간 순서에 따라 보여지며 다이어그램의 수직방향이 시간의 흐름을 나타낸다.
- 객체는 다른 객체와 메시지를 주고받는다.
- 각 메시지는 시간의 흐름에 따라 순서를 정하게 된다.

● 메세지

- 한 객체에서 다른 객체로 전송되는 메시지는 한 객체의 생명선에서 다른 객체의 생명선으로 이동하는 것을 의미
- 화살표로 메시지를 표현하는데, 화살표의 머리 모양이 메시지의 형태를 좌우
- 호출(Call) 메시지
 - ✓ 실선의 끝에 속을 칠한 화살표([그림 5-6] (a))
- 답신 메시지
 - ✓ 점선 끝에 화살표([그림 5-6] (b))를 붙인 모양
 - ✓ 수신 객체로부터 답신 메시지를 요청하는 경우, 답신 메시지의 표시를 생략
- 동기(Synchronous) 메시지
 - ✓ 송신 객체가 수신 객체를 기다려주는 메시지
- 비동기(Asynchronous) 메시지
 - ✓ 송신 객체가 보내는 메시지로서 수신 객체의 오퍼레이션을 실행하게 하는 것은 동일하지만, 오퍼레이션이 완료될 때까지 송신 객체가 기다리지는 않는다.
 - ✓ [그림 5-7]과 같이 실선에 화살표모양을 붙여 나타낸다



(a) 호출(동기) 메시지

[그림 5-6] 동기 메시지

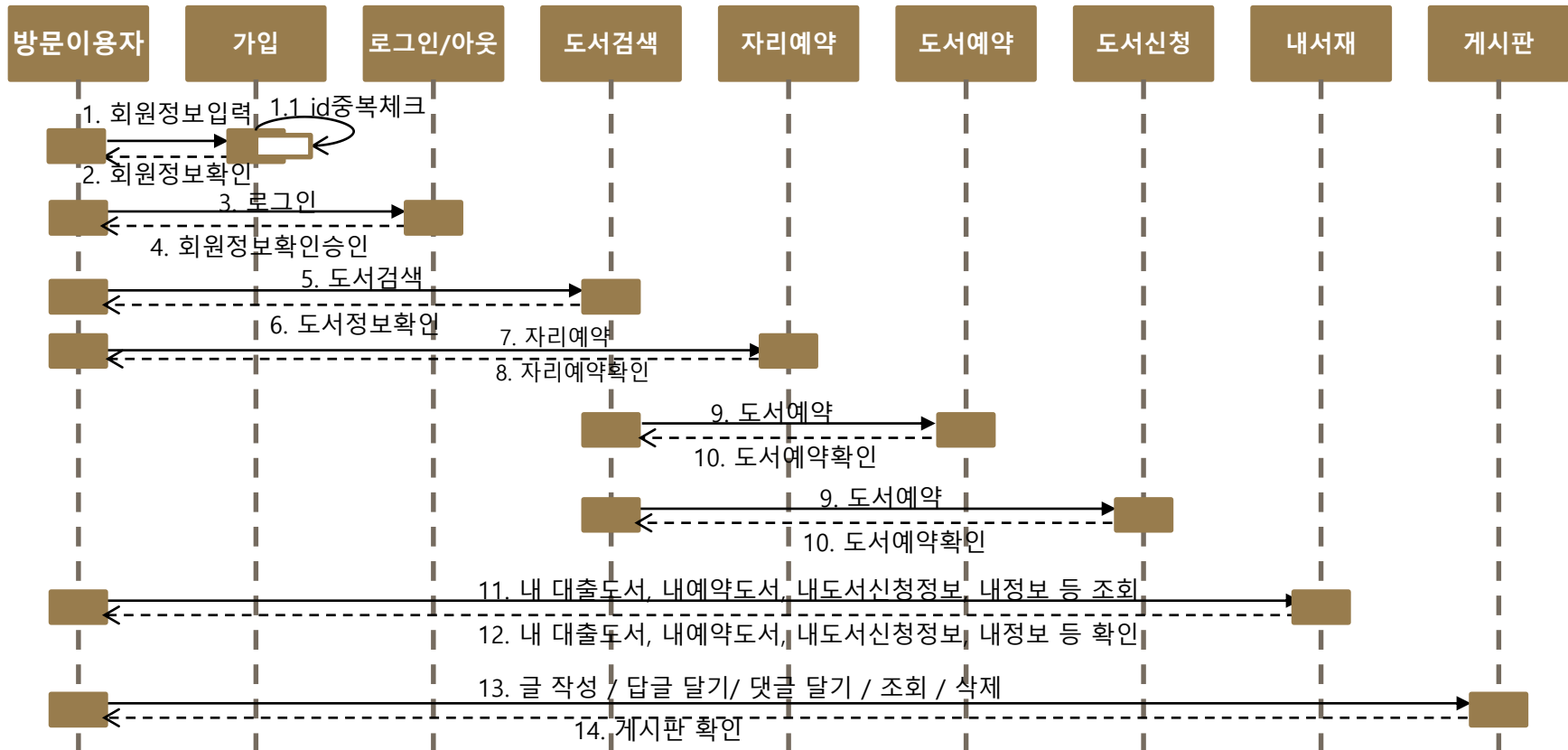


(b) 답신 메시지



[그림 5-7] 비동기 메시지

도서관 시스템의 순차 다이어그램 예



1

연계 메커니즘 구성하기

≡ 연계 구조 구성하기

2-1 연계 매커니즘 정의

<표 2-2> 연계 방식 분류

방식	연계 방식별 상세 설명	예시	송수신 서버(엔진)
DB Link	<ul style="list-style-type: none"> - 데이터베이스에서 제공하는 객체(Object) 이용 - 수신 시스템 디비(DB)에 송신 시스템에서 접근 가능한 디비링크(DBLINK: Database Link) 객체를 생성한 후 송신 시스템에서 디비링크명(DBLINK명)으로 직접 참조하여 연계 	테이블 명 @DBLink명	X
직접 연계 DB Connection Pool (WAS)	<ul style="list-style-type: none"> - 수신 시스템 와스(WAS)에서 송신 시스템 DB로 연결되는 커넥션 풀(Connection Pool) 생성 - 프로그램 소스에서 WAS에 설정된 Connection Pool명을 참고하여 구현 	송신 시스템 데이터 소스(DS: Data Source) (WAS에서 설정한 Connection Pool명)	X
JDBC	<ul style="list-style-type: none"> - 수신 시스템의 배치(Batch) 또는 온라인(On-Line) 프로그램에서 제이 디 비 씨(JDBC) 드라이버를 이용하여 송신 시스템의 DB와 연결 생성 	jdbc:DBMS 제품명: DBMS 설치 서버 IP: Port:DB instance명	X
화면 링크 (Link)	<ul style="list-style-type: none"> - 웹 애플리케이션 화면에서 하이퍼 링크(Hyper Link) 이용 	 LINK 화면명	X
API 또는 Open API	<ul style="list-style-type: none"> - 송신 시스템의 DB와 연결하여 데이터 제공하는 인터페이스(Interface) 프로그램 	제공하는 컴포넌트명. 데이터 처리 메서드명 (파라미터 1-N);	△
연계 솔루션 (EAI)	<ul style="list-style-type: none"> - 실제 송수신 처리와 진행 현황을 모니터링 및 통제하는 이 에이 아이(EAI) 서버, 송수신 시스템에 설치되는 Adapter(Client)를 이용 	메타빌드, 비즈마스터(Biz Master) 등	○
간접 연계 Web Service/ ESB	<ul style="list-style-type: none"> - 웹 서비스가 설명된 더블유 에스 디 엘(WSDL)과 숀(SOAP) 프로토콜을 이용한 시스템 간 연계 - 미들웨어인 이 에스 비(ESB)에서 서비스(컴포넌트) 간 연동을 위한 변환 처리로 다중 플랫폼(Platform) 지원 	WSDL, 유 디 디 아이(UDDI), 숀(SOAP), 서비스(Service), ESB(Enterprise Service Bus)	○

Framework

- 프레임워크란 목적에 따라 효율적으로 프로그램의 구조를 짜놓고 개발하는 방식
- 구조에 맞는 다양한 api가 제공되고 개발 효율성과 유지보수가 편리해진다
- 장고, 플라스크, 스프링, 앵귤러js, 안드로이드 프레임워크 등이 그 예이다

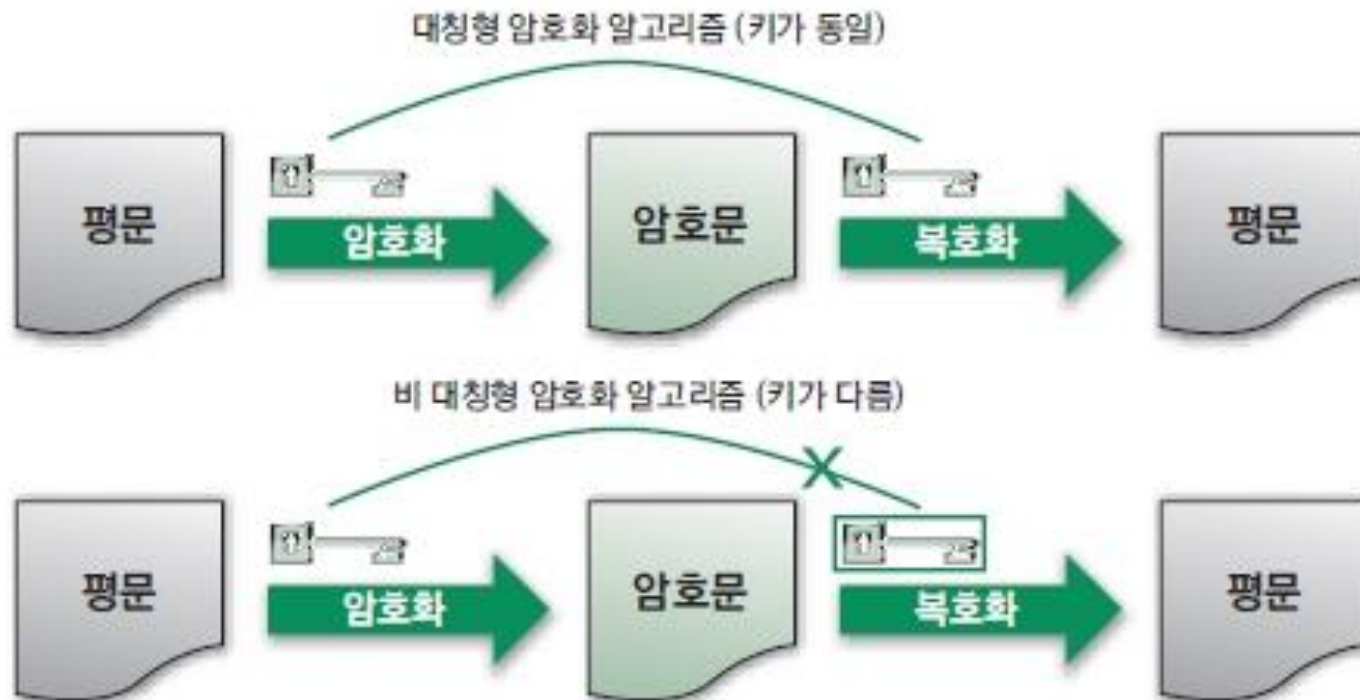
2-2 연계 장애 및 오류 처리 구현

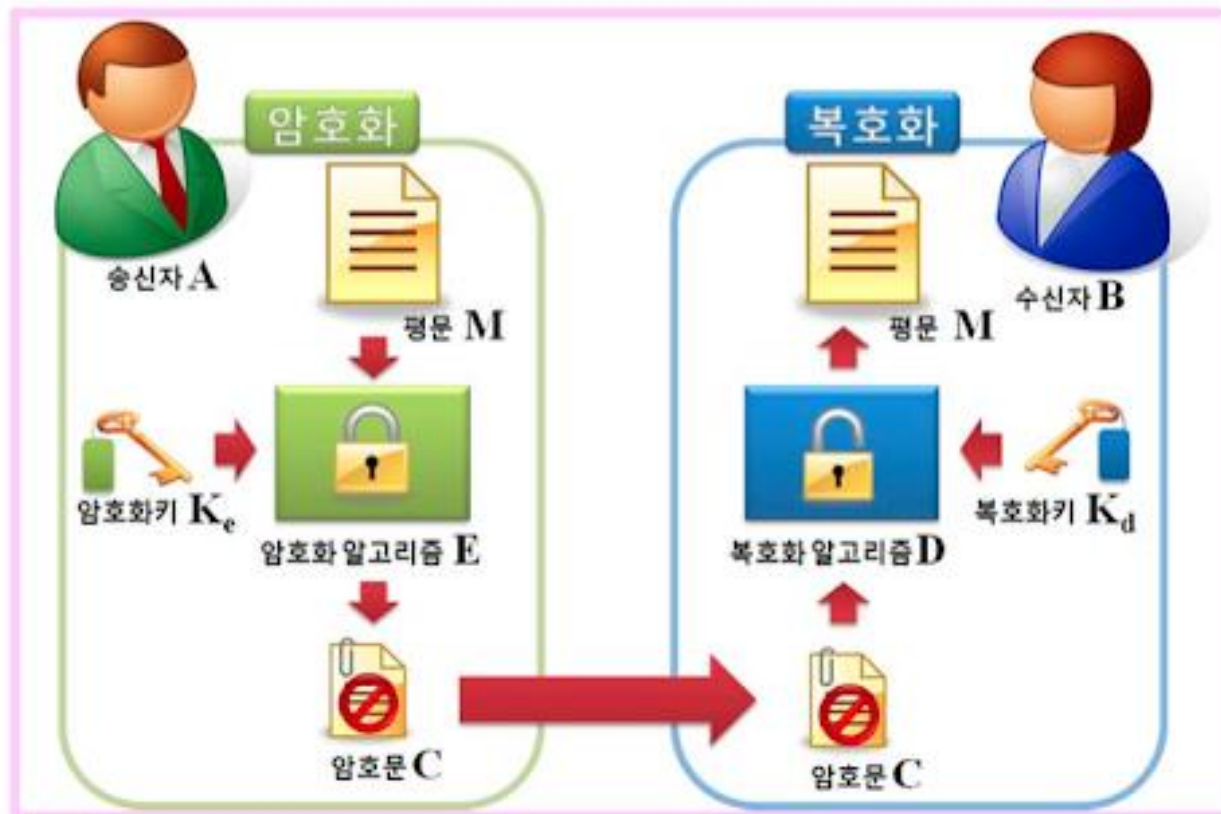
<표 2-13> 장애 및 오류 유형

오류 유형	설명	사례	처리 방안
연계 시스템	연계 서버의 실행 여부를 비롯하여 송수신, 전송 형식 변환 등 서버의 기능과 관련된 장애 및 오류	연계 서버 미기동, 송수신 시스템의 아이피(IP) 및 포트(Port) 접속 불가	연계 서버(엔진)의 로그를 확인하여 원인 분석 후 처리
연계 응용 프로그램	송신 시스템 연계 프로그램 데이터 추출을 위한 DB 접근 권한 불충분, 데이터 변환 시 익셉션(Exception) 미처리 등 연계 프로그램 구현상의 오류	등록되지 않은 코드로 매핑 불가	송수신 시스템 연계 프로그램에서 기록하는 로그(Log)를 확인하여 원인 분석 후 결과에 따른 처리, 처리 이후 데이터 전송이나 반영 재작업
	수신 시스템 연계 프로그램 운영 DB에 반영하기 위한 DB 접근 권한 불충분, 데이터 변환 및 반영 시 익셉션(Exception) 미처리 등 연계 프로그램 구현상의 오류	등록된 데이터가 존재하지 않음 ⇒ 등록되지 않은 데이터에 대한 수정(Update) 처리	
연계 데이터 오류	송신 시스템에서 추출된 연계 데이터가 유효하지 않은 값으로 인한 오류	유효하지 않은 일자	송신 시스템 연계 프로그램에서 기록하는 Log를 확인하여 데이터 보정 후 재전송

2-3 연계 데이터 보안 적용


ex. 개인식별정보나 금융정보





연계 메커니즘에서 데이터 암호화 복호화 절차

개인식별정보나 금융정보가 들어 있는 암호화 적용 대상 데이터, 알고리즘, 암호화 키등을 결정한다.



암호화 알고리즘 라이브러리(API) 확보 및 설치



연계 응용 프로그램에서 암호화 및 복호화 로직을 처리한다