

# 객체지향 가상상점 프레임워크 설계 및 구현

김 동 관<sup>o</sup>, 이 상 덕, 양 영 종, 정 효 택

{dgkim, lsd, yangyj, htjung}@etri.re.kr

한국전자통신연구원 컴소연 컴포넌트기반 SW 연구팀

## A Design and Implementation of Object-Oriented Virtual Shopping Mall Frameworks

Dong-Kwan Kim, Sang-Duck Lee, Young-Jong Yang, Hyo-Taeg Jung

Component-based Software Research Team, Real-time Computing Dept., CSTL, ETRI

### 요 약

소프트웨어 개발 시 기존 전문가들의 개발 경험을 재사용하는 작업은 개발 기간의 단축 및 비용의 절감의 효과를 가져올 수 있다. 이러한 재사용에 관련된 연구는 소프트웨어공학 분야에서 계속되고 있으며 그 하나로 객체지향 프레임워크를 들 수 있다. 객체지향 프레임워크는 검증된 소프트웨어 재사용 부품으로 특정 도메인에 대한 비즈니스 로직을 재사용할 수 있도록 구현된다. 프레임워크 내부에는 재사용되는 로직이 포함되어 있으며 프레임워크 사용자에게 커스터마이징 될 모듈을 정의하고 있다. 프레임워크 사용자는 프레임워크를 기반으로 어플리케이션을 개발함으로써 처음부터 개발을 시작하지 않으며 기 개발자들의 경험을 재사용하게 된다. 본 논문에서는 근래 크게 대두되고 있는 가상상점 도메인에 대한 프레임워크를 제시한다. 본 논문에서 제시하는 가상상점 프레임워크는 메인 프레임워크와 주문, 지불, 배달의 3개 서브프레임워크로 구성되며 각각의 프레임워크가 포함한 핫스팟, 제어흐름, 프레임워크 인터페이스를 중심으로 프레임워크의 구성요소 및 설계, 구현에 관한 사항을 기술한다.

### 1. 서론

객체지향 프레임워크 [1][2][3]는 특정 도메인의 공통 기능을 미리 구현한 프로그램으로 소스 수준에서의 대규모 재사용을 지원한다. 프레임워크는 핫스팟(hot spot), 제어흐름(control flow), 프레임워크 인터페이스(interface)로 구성된다. 핫스팟은 프레임워크를 확장할 수 있는 부분으로 클래스나 메소드 형태로 정의되며 프레임워크 사용자는 핫스팟을 커스터마이징함으로써 원하는 어플리케이션을 개발할 수 있다. 제어흐름은 프레임워크 내부에 기 정의된 로직을 말한다. 예를 들면, 본 논문에서 제시한 가상상점 프레임워크 가운데 주문 프레임워크의 경우, 주문 신청, 주문 변경, 주문 취소 등의 비즈니스 로직에 해당하는 제어흐름을 가진다. 프레임워크의 재사용은 기 정의된 제어흐름의 재사용을 의미한다. 제어흐름은 일련의 관련된 메시지들의 순서이다. 프레임워크 인터페이스는 프레임워크와 프레임워크 외부의 클래스나 다른 프레임워크를 연결하는 통로로 프레임워크 사용자는 프레임워크 인터페이스를 통해 프레임워크에 정의된 제어흐름을 재사용한다. 핫스팟, 제어흐름, 프레임워크 인터페이스는 상호 유기적이며 프레임워크의 품질을 평가하는 기준으로 활용될 수 있다.

본 논문은 요즘 이슈가 되고 있는 전자상거래 시스템 개발에 활용될 수 있는 가상상점 프레임워크의 설계 및 구현 사례를 제시한다. 본 논문에서 제시한 가상상점 프레임워크를 기반으로 가상상점 어플리케이션을 개발함으로써 개발기간 및 비용의 절감 효과를 기대할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 가상상점 프레임워크의 핵심이 되는 주문, 지불, 배달 프레임워크의 정적 및 동적 측면을 살펴보고 3장에서는 2장에서 기술한 가상상점 프레임워크를 기반으로 개발된 어플리케이션을 기술한다. 4장은 결론 및 향후연구과제를 기술한다.

### 2. 가상상점 프레임워크

본 논문의 다이어그램들은 UML(Unified Modeling Language) [4] 표기법을 따른다. 가상상점 도메인의 업무흐름은 다음과 같다. 고객이 로그인을 하고 로그인이 정상적으로 되면 상품을 검색한다. 검색과정에서 상품들을 주문하고 주문된 상품에 대해 상품 구매를 하게 된다. 상품 구매가 이루어지면 주문에 대한 지불 행위가 이루어지게 되며 상품 구매에 대한 취소도 가능하다. 그리고 상품에 대한 지불이 완료되면 상품 배달이 진행된다. 상품 배달이 정상적으로 진행되면 업무흐름이 종료되며, 배달된 상품에 손상이나 배달이 지연될 경우에는 상품 반환이 발생한다.

가상상점 프레임워크는 메인 프레임워크, 주문 프레임워크, 지불 프레임워크, 배달 프레임워크로 구성되며 메인 프레임워크는 프레임워크에서 공통적으로 사용하는 프로퍼티(property)들을 관리하는 기능을 수행한다. 시스템 프로퍼티에는 데이터베이스 관리자 설정, JDBC (Java DataBase Connectivity) [5] 드라이버 설정, 배달 유형 등이 포함된다. 핵심이 되는 3개의 프레임워크에 대해서는 아래의 장에서 기술한다.

#### 2.1. 주문 프레임워크

주문 기능을 수행하는 서브 프레임워크로 고객은 주문 프레임워크에서 제공하는 쇼핑리스트에 주문하려는 상품 내역을 삽입하고 주문을 수행한다.

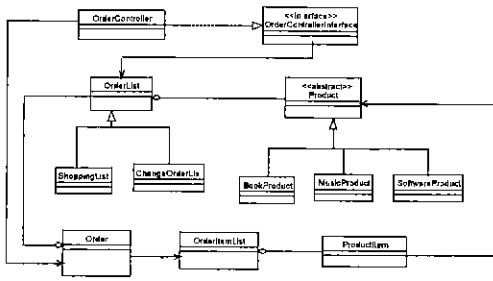


그림 1 주문 프레임워크의 클래스 다이어그램

그림 1은 주문 프레임워크에 대한 간략화된 클래스 다이어그램을 보여준다. 주문 프레임워크의 프레임워크 인터페이스 기능은 OrderControllerInterface 클래스가 담당한다. OrderControllerInterface의 메소드에는 processOrder(), changeOrder(), cancelOrder() 등이 있다. OrderController는 OrderControllerInterface에서 정의한 메소드들을 정의하고 있으며 실질적인 처리는 해당 클래스에서 이루어진다. 주문 프레임워크의 핫스팟은 Product클래스로 프레임워크 사용자는 Product클래스를 상속 받아 특정 상품을 정의할 수 있다. 그림 1에서는 책, 음악 CD, 소프트웨어를 예로 제시한다.

주문 프레임워크에서 재사용되는 대표적인 제어흐름은 주문 처리, 주문변경, 주문취소이며 그림 2에서는 주문처리에 대한 제어흐름을 보여준다. 프레임워크 외부는 프레임워크 인터페이스인 OrderControllerInterface클래스를 통해 프레임워크에 메시지를 전달하고 결과값을 받는다. 고객의 입력에 의해 ShoppingList가 생성되고 생성된 ShoppingList에는 상품에 대한 추가, 삭제 등의 작업이 수행된다. ShoppingList에 원하는 상품이 추가되면 고객은 주문처리를 요구한다. 주문처리는 프레임워크에 기 정의된 제어흐름에 따라 처리된다. 일단 상품재고 여부를 판단하고 가용하면 재고를 변경한 후 주문 현황을 저장한다.

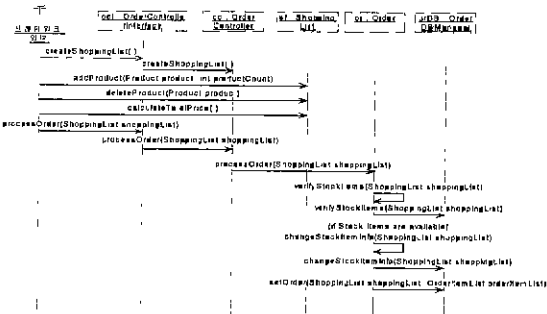


그림 2 주문처리에 대한 제어흐름

## 2.2. 지불 프레임워크

지불 프레임워크는 지불 기능을 수행하는 서버 프레임워크로 주문 프레임워크 기능을 통해 주문이 성공적으로 발생한 경우에 사용할 수 있다. 고객의 주문 내역과 지불 정보를 결제 기관으로 전송하여 결제 기관에 결제 처리를 의뢰한다. 고객의 신용도가 불량하거나 고객의 결제 수단이 유효하지 않을 경우는 결제 처리가 되지 않는다.

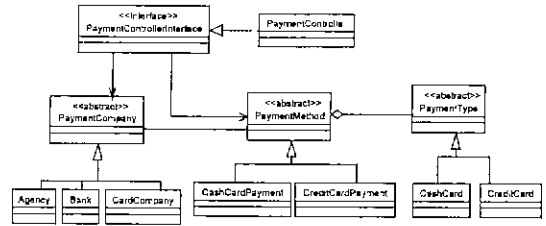


그림 3 지불 프레임워크의 클래스 다이어그램

그림 3은 지불 프레임워크에 대한 간략한 클래스 다이어그램을 나타낸다. 지불 프레임워크에 대한 외부 접근은 프레임워크 인터페이스인 PaymentControllerInterface에 의해 이루어지며 PaymentController는 프레임워크 인터페이스를 구현한다. PaymentControllerInterface클래스에는 processPayment(), cancelPayment(), requestAuthorization(), requestBalanceCheck() 등의 지불처리와 관련된 메소드들이 정의된다. 지불회사, 지불방법, 지불유형은 추상클래스로 정의되며 각각 구체적 클래스에 의해 상속된다. 해당 클래스들은 어플리케이션에 따라 달라질 수 있으므로 핫스팟으로 처리된다.

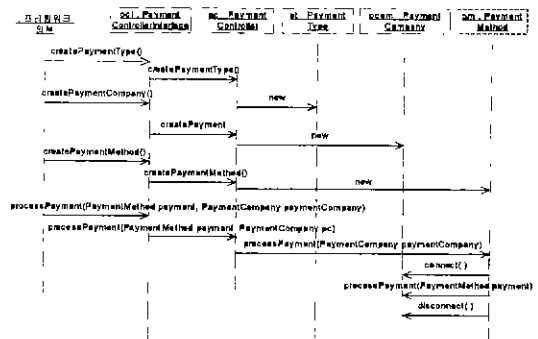


그림 4 지불 처리에 대한 제어흐름

그림 4는 지불 프레임워크에서의 지불 처리 제어흐름을 보여준다. 고객에 요구에 의해 PaymentType, PaymentCompany, PaymentMethod 객체가 생성되며 PaymentControllerInterface 프레임워크 인터페이스의 processPayment() 메소드를 호출함으로써 지불 처리가 수행된다. 지불 프레임워크는 외부 시스템과의 연동이 요구되며 실질적인 지불 처리는 지불회사에서 이루어지므로 PaymentCompany클래스를 상속한 지불회사 클래스를 구현하여 사용해야 한다.

## 2.3. 배달 프레임워크

배달 프레임워크는 상품 배달 기능을 수행하는 서버 프레임워크로 배달 방법을 설정하고 해당 배달 회사에게 배달을 의뢰하는 기능을 제공한다.

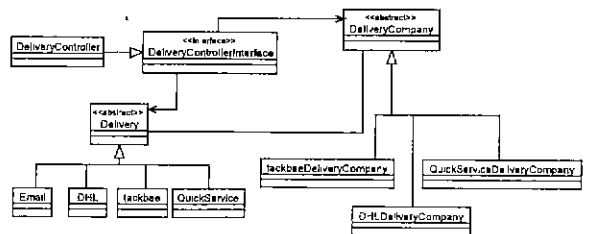


그림 5 배달 프레임워크의 클래스 다이어그램

그림 5은 배달 프레임워크에 대한 간단한 클래스 다이어그램이다. 핫스팟에 해당하는 클래스는 DeliveryCompany와 Delivery클래스이다. Delivery클래스는 배달방식에 대한 최상위 클래스로 배달회사에게 배달을 요청하고 배달에 관한 사항을 데이터베이스에 저장하는 기능을 수행한다. 본 배달 프레임워크에서는 QuickService클래스를 디폴로 정의한다. 배달 제어 흐름은 배달 등록, 배달 완료, 배달 취소 등으로 구분된다. 일단 배달 등록이 수행되면 해당 주문은 배달 중인 상태가 되고 배달이 완료되면 배달 완료 상태가 저장된다. 사용자의 요구에 따라 배달 취소가 발생할 수도 있다.

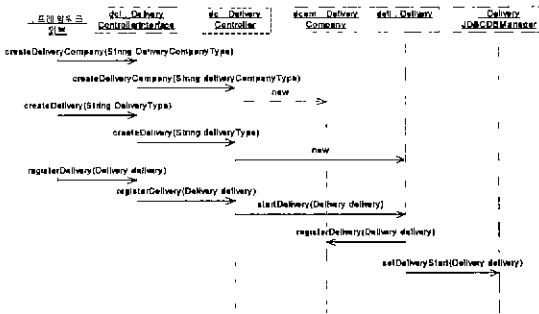


그림 6 배달 등록에 대한 제어흐름

그림 6은 배달 등록에 대한 제어흐름을 나타낸다. 프레임워크 외부에서 DeliveryControllerInterface의 registerDelivery() 메소드가 호출됨으로써 배달 등록은 사전에 정의된 흐름에 따라 처리된다. Delivery클래스의 startDelivery() 메소드가 호출되며 Delivery클래스는 DeliveryCompany클래스의 registerDelivery() 메소드를 호출한 후 DeliveryJDBCDBManager클래스의 setDeliveryStart()메소드를 호출한다. 실질적인 배달 등록 처리는 배달회사에서 이루어지므로 Delivery클래스의 registerDelivery()에 의해 처리된다. setDeliveryStart()메소드는 특정 주문이 "배달 중"임을 데이터베이스에 저장한다.

3. 가상상점 어플리케이션

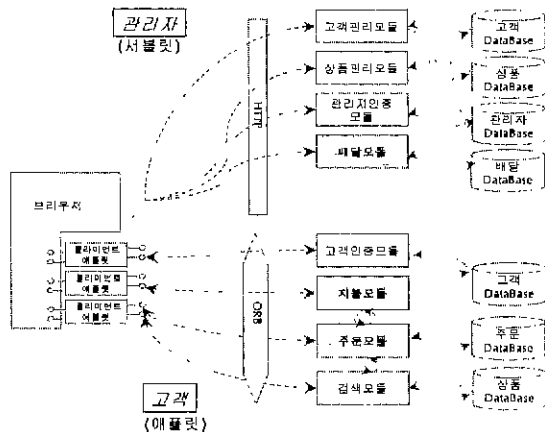


그림 7 가상상점 어플리케이션 구성도

그림 7은 가상상점 프레임워크를 기반으로 개발된 가상상점 어플리케이션 시스템의 구성도를 나타낸다. 그림에서 배달모듈,

지불 모듈, 주문 모듈은 가상상점 프레임워크를 나타낸다. 나머지 5개의 모듈은 새롭게 추가된 기능이다. 프레임워크의 범위를 어디까지 정하냐에 따라 5개의 모듈 가운데 프레임워크화할 수 있는 부분도 생길 수 있다.

그림 7의 어플리케이션은 고객 모드와 관리자 모드로 나뉜다. 고객은 시스템에 접근하여 상품 구매 행위를 수행하며 관리자는 상품 및 고객, 관리자에 대한 정보를 관리한다. 프레임워크와 어플리케이션은 Java[5] 언어로 구현되었으며 고객은 웹 브라우저를 사용해 해당 시스템에 접근하며 애플릿[5]을 통해서 서버와 통신한다. 미들웨어(middleware)는 CORBA(Common Object Request Broker Architecture) [6]로 구현된다. 관리자 모드는 서블릿(servlet) [5]을 사용해 구현한다.

4. 결론 및 향후 연구과제

본 논문에서는 가상상점 어플리케이션 개발 시 개발 기간 단축 및 비용 절감을 효과를 가져올 수 있는 객체지향 기반의 가상상점 프레임워크를 제시하고 있다. 가상상점 프레임워크는 메인 프레임워크와 주문, 지불, 배달의 3개의 서브 프레임워크로 구성된다. 메인 프레임워크는 서브 프레임워크에 공통적으로 요구되는 데이터베이스 등의 시스템 환경을 설정하는 기능을 수행하고 서브 프레임워크는 비즈니스 로직에 대한 제어흐름을 구현한다. 주문 프레임워크는 주문신청, 주문취소, 주문변경 등의 제어흐름을, 지불 프레임워크는 지불과 관련된 지불처리, 광고확인, 신용도확인 등의 제어흐름을, 배달 프레임워크는 배달등록, 배달취소, 배달시작, 배달확인 등의 제어흐름을 재사용할 수 있다. 서브 프레임워크는 프레임워크 사용자들이 확장할 수 있는 핫스팟을 제공한다.

향후연구과제로 가상상점 프레임워크와 배달회사, 지불처리 회사 등의 외부 시스템과의 연동 기능에 대한 연구가 요구된다. 또한, 어플리케이션 개발 기간 단축 및 개발의 용이성을 위해 프레임워크기반 어플리케이션 개발 방법론 및 이를 지원하는 도구에 대한 연구가 필요하다.

참 고 문 헌

[1] Wolfgang Pree, Design Patterns for Object-Oriented Software Development, Addison-Wesley, 1994.  
 [2] Niklas Landin, Axel Niklasson, "Development of Object-Oriented Frameworks", Master Thesis, Department of Communication Systems, Lund Univ., Sweden, 1995.  
 [3] Michael Mattson, "Object-Oriented Frameworks A survey of methodological issues", Department of Communication Systems, Lund Univ., Sweden, 1996.  
 [4] Martin Fowler, Kendall Scott, UML Distilled Applying the Standard Object Modeling Language, Addison-Wesley, 1997  
 [5] <http://www.javasoft.com>  
 [6] <http://www.omg.org/corba>