B2C 쇼핑몰 시스템을 위한 프로덕트 라인

조영호° 윤병권 최윤석 정기원*
숭실대학교 컴퓨터학과, 숭실대학교 컴퓨터학부*
snatcher93@hanmail.net, superman@it.soongsil.ac.kr, yschoi@it.soongsil.ac.kr, chong@computing.soongsil.ac.kr

A Product Line for B2C Shopping Mall Systems

Youngho Cho° Byungkwon Yoon Younseok Choi Kiwon Chong*
Dept. of Computing, Soongsil Univ. School of Computing, Soongsil Univ.

요 약

B2C 쇼핑몰 시스템의 경우 사용자 인터페이스에 관련된 부분에서만 차이를 보일 뿐 시스템의 기능적인 요구사항과 아키텍처는 유사하며 재사용 가능한 컴포넌트나 COTS 제품을 사용하여 시스템의 많은 부분을 구축할 수 있다. 따라서 쇼핑몰 시스템 구축 시 매번 각 기능들을 설계하고 구현하는 짓은 중복 투자를 하는 것과 같다. 이에 본 논문에서는 B2C 쇼핑몰 시스템의 공통 요구사항을 기반으로 B2C 쇼핑몰 시스템 구축을 위한 프로덕트 라인을 제시한다. 쇼핑몰 시스템 구축에 프로덕트 라인을 적용한다면 중복 투자를 줄이고 미리 준비한 COTS 제품이나 컴포넌트를 사용하여 좋은 품질의 시스템을 빠르게 구축할 수 있다. B2C 쇼핑몰 시스템의 프로덕트 라인을 제시하기 위해 프로덕트 라인의 Core Asset Development 프로세스를 수행하여 쇼핑몰 시스템에 특화된 Product Line Scope, Core Assets, Production Plan을 정의하였다. B2C 쇼핑몰 시스템의 Product Line Scope는 웹 상에서 물건 판매, 주문 정보 접수, 고객 정보 관리 등의 공통점과 가격의 고정 여부, 검색 등 기타 기능, 공동구매 기능 등의 차이점으로 정의할 수 있다. Core Asset은 B2C 쇼핑몰 시스템의 아키텍처와 시험 및 설계 방법, 재사용 가능한 컴포넌트 목록 등이 있다. 마지막으로 Production Plan은 각각의 Core Asset의 부착 프로세스를 연결하여 제시한다. B2C 쇼핑몰 시스템의 Production Plan은 프로젝트 계획, 아키텍처 정제, 사용자 인터페이스 프로토타이핑, 컴포넌트 획득, 구현, 통합 및 테스팅, 운영 단계로 구성된다.

1. 서론

대부분의 B2C 쇼핑몰 시스템은 컨텐츠나 판매 방식 등의 사용자 인터페이스에서만 차이를 보일 뿐 고객 관리, 상품 정보 관리, 주문 정보 접수, 상품 진열 등의 많은 기능적/비기능적 요구사항과 아키텍처는 동일하여 재사용이 가능하다. 따라서 쇼핑몰 시스템의 공통적인 요구사항과 기본 아키텍처를 정의하고 이를 재사용 한다면 다른 쇼핑몰 시스템을 구축할 때 설계나 구현에 따른 중복 투자를 막을 수 있으며 시스템 구축의 노하우와 COTS 제품이나 이전의 시스템 구축에 사용했던 컴포넌트를 재사용해 고품질의 시스템을 빠르게 구축할 수 있어 보다 많은 수익을 올릴 수 있다.

그러나 기존의 개발방식은 특정 고객을 위주로 한 프로젝트의 관리방식이었기 때문에 응용마다 별개로 관리되어져왔다. 대부분의 개발업체들은 특정 도메인의 유사한 응용을 개발하기 마련인데, 기존의 방법은 초기 개발 시에 재사용을 염두하고 있지 않기 때문에, 추후에 유사한 프로젝트를 수행하면서도 이전에 개발한 요소를 재사용하기가 매우 어려운 실정이다. SEI에서는 언급된 문제점들을 지적하고 프로덕트 라인 프레임웍이라는 새로운 접근방식을 제안하였다 [4].

프로덕트 라인이란 특정 시장 또는 임무의 요구를 만족시키는 공통적이고 관리된 특성을 공유하는 제품의 집합이다.[5] 즉, 공통 특성을 공유하는 프로덕트 패밀리를 정의하여 아키텍처, 컴포넌트, 인터페이스등을 공유하며 이를 통해 제품의 생산성을 높일 수 있는 방법이다.

본 논문에서는 B2C 쇼핑몰 시스템의 공통 요구사항을 기반으로 B2C 쇼핑몰 시스템 구축을 위한 프로덕트 라인을 제시한다. 일반적으로 프로덕트 라인을 적용시킨 경우 마케팅 시간이 절약되며 대략적으로 10배의 생산성 향상과 60% 정도의 비용이 절감된다. 또한 제품 생산에 필요한 노동력이 1/10로 감축되고 제품의 질은 10배정도 향상된다[8].

2. 관련연구

2.1 쇼핑몰 시스템

B2C 쇼핑몰 시스템의 비즈니스 모델은 Storefront 모델, Auction 모델, Portal 모델, Dynamic Pricing 모델로 분류할 수 있다.[2] 본 논문에서는 위에 열거한 모델 중 Storefront 모델과 Dynamic Pricing 모델에 초점을 둔다.

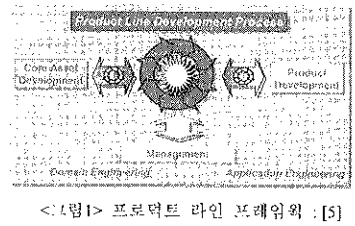
Storefront 모델은 가장 일반적인 인터넷 비즈니스 모델로서 웹 상에서 물건의 카탈로그를 나열하고 그에 따른 주문정보를 받을 수 있어야 하며, 상품을 고객에게 운송하고 고객 정보를 관리할 수 있어야 한다. Dynamic Pricing Model은 웹 상에서의 상품 판매를 목적으로 한다는 점에서는 Storefront 모델과 동일하지만 상품의 가격이 유동적이라는 점에서 차이가 있다. B2C 쇼핑몰 시스템에서 개발하고자 하는 웹 어플리케이션은 적어도 클라이언트 브라우저, 웹 서버, 어플리케이션 서버의 세 가지 중요한 아키텍처 컴포넌트를 가진다. 현재 존재하는 여러 종류의 웹 어플리케이션 아키텍처 패턴 중에서 가장 중요한 세 가지 패턴은 Thin Web Client, Thick Web Client, Web Delivery이다[1]. 전자상거래를 위한 어플리케이션 프레임워크는 데이터, 연결성, 네트워크 인프라, 그리고 어플리케이션 소프트웨어 컴포넌트들에 대한 참조가 필요하다. 만일 컴포넌트들이 상호운용이 불가능하여 함께 사용할 수 없다면 시스템 구축에 적합하지 않다. 이러한 문제를 해결하기 위해 전자상거래를 위한 객체지향 프레임워크가 제시되었다. IBM, 넷스케이프, 오라클, 그리고 SUN과 같은 업체들은 이를 위해 CORBA IIOP에 기반한 공통 분산 객체 모델인 eCo 아키텍처를 지원하기로 동의했으며[3] Web Delivery 패턴이 eCo 아키텍처를 수용할 수 있다.

2.2 프로덕트 라인

프로덕트 라인이란 특정 시장 또는 임무의 요구를 만족시키는 공통적이고 관리된 특성을 공유하는 제품의 집합이다. 프로덕트 라인에 의해 개발될 새로운 제품은 이미 확보되었거나 새로이 개발된 컴포넌트들의 조합으로 구성된다[5]. 프로덕트 라인 프레임웍을 구축하는 활동을 Core Asset Development 프로세스라고 하고, 구축된 프레임웍을 통해 제품을 작성하는 활동을 Product Development 프로세스라고 한다[4]. <그림 1>에서 보는 바와 같이 각 단계는 반복적이며 하나의 단계에 대한 산출물이 다른 단계에 대한 입력으로 사용될 수 있다. 프로덕트 라인 프레임웍 구축의 핵심을 이루는 Core Asset Development 프로세스는 Product Line Scope, Core Assets, Production Plan을 정의하는 프로세스이다.

2.3 프로덕트 라인의 B2C 쇼핑몰 시스템 적용시 장점

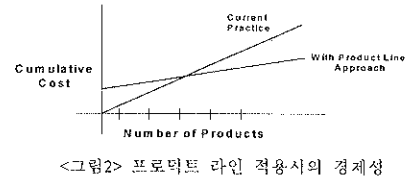
소프트웨어 프로덕트 라인은 전략적인 재사용을 의미한다. 따라서 B2C 쇼핑몰 시스템에 프로덕트 라인을 적용할 경우 제품간의 공통점을 효율적으로 관리할 수 있고, 융통성과 통제성 간의 균형을 유지할 수 있으며, 상세한 프로세스와 경량 프로세스 간의 균형을 유지할 수 있다.



<그림1> 프로덕트 라인 프레임웍 : [5]

상업적인 이익으로는 제품 개발 기간이 단축되고, 후에 증가 비용이 적게 소요되므로 개발 금액이 감소하며, 제품의 질이 높아지고, 제품의 경쟁력을 갖출 수 있다.

<그림 2>는 Lucent Technologies Bill Laboratories Innovation에서 제공하는 데이터를 바탕으로 추출한 것으로 일반적인 방법론에 대해 프로덕트 라인을 적용할 경우의 경제성을 나타낸다. 일반적인 방법론을 적용한 경우 초기 개발 비용은 적게 들지만 제품을 개발할 때마다 추가되는 비용이 많은 반면 프로덕트 라인 적용시에는 초기 개발 비용이 많이 드는 반면 제품 개발 시의 추가 비용이 적다는 것을 알 수 있다.

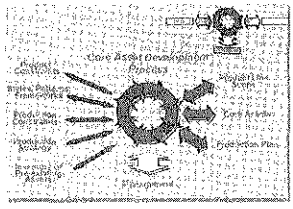


<그림2> 프로덕트 라인 적용시의 경제성

3. B2C 쇼핑몰 시스템 프로덕트 라인 모델

본 논문에서 제시하는 모델은 SEI에서 제시한 기본 프로덕트 라인의 Core Asset Development 프로세스를 B2C 쇼핑몰 시스템 구축에 적용하는 방법에 초점을 맞추고 있다.

<그림 3>은 SEI에서 제시한 기본 프로덕트 라인의 Core Asset Development 프로세스를 나타내고 있다. 프로덕트 라인의 Core Asset Development 프로세스는 1)Product Constraints, 2)Styles, Patterns, and Frames, 3)Production Constraints, 4)Production Strategy, 5)Inventory of Preexisting Asscts의 5가지 입력물을 반복적인 프로세스로 처리하여 Product Line Scope, Core Assets, Production Plan을 정의한다. 또한 양방향 화살표가 의미하는 바대로 각각의 입력물은 반복적인 프로세스를 통한 피드백에 의해 정제가 가능하다[5].



<그림 3> Core Asset Development : [5]

3.1 Core Asset을 개발하기 위한 입력물

2.1절에서 기술한 비즈니스 모델간의 차이점을 바탕으로 프로덕트 라인을 구성하는 공통점과 차이점을 식별하여 Product Constraints를 정의한다. 또한 2.1절에서 언급한 바와 같이 eCo 아키텍처를 수용할 수 있는 Web Delivery 패턴을 적용하여 Styles, Patterns, and Frameworks를 정의한다.

표1. B2C 쇼핑몰 시스템 구축 시 사용 가능한 소프트웨어 컴포넌트의 예

Parts	Categories	Products
Front Shop	Client & Clerk	Internet Explorer, Navigator, Mosaic
Back Office	Operating System	*Windows NT*, Windows 2000, UNIX, LINUX
	Web Server	IIS, Apache
	Application Server	MTS, WebLogic, *Oracle Application Server*, Enterprise Application Server, Netscape Application Server, Inprise Application Server
	Database Server	Oracle, DB2, Sybase, Infomix, Ingres, MS SQL Server
	ORB	Java IDL, OrbixWeb, Visibroker
	DB Connectivity	ODBC, JDBC
	Payment Gateway	XPAY Gateway 2.0, Credit Payment Gateway 2.0, Globeset Gateway 1.4, IBM Payment Gateway 1.2[6]

Production Constrains는 제품이 만들어질 기반과 사용할 COTS, 재사용할 기존 컴포넌트의 목록을 정의한다. 시스템에 사용할 컴포넌트를 선정하기에 앞서 우선 쇼핑몰 시스템이 운영될 플랫폼에 대한 고려가 있어야 한다[6]. 플랫폼 선정 후에는 표1과 같이 사용 가능한 소프트웨어 컴포넌트의 목록을 작성한 후 COTS와 재사용할 수 있는 컴포넌트의 목록을 추출할 수 있다[6].

Production Strategy는 B2C 쇼핑몰 시스템 개발을 위한 관련 컴포넌트와 컴포넌트의 개선을 위한 정로를 규정한다. 일반적으로 제품 개발 시에 Top-Down 방식을 적용할 것인지, *Bottom-Up* 방식을 적용할 것인지를 결정한다. 본 논문에서는 제품 개발에 필요한 Core Asset을 정의한 후 이를 통합하여 제품을 개발하는 Top-Down 프로덕트 라인 모델을 제시할 것이다. Inventory of Preexisting Assets은 이미 확보된 컴포넌트를 나열하며 이를 바탕으로 제품 개발 시에 적절한 컴포넌트를 선정할 수 있다.

3.2 Product Line Scope

Product Line Scope를 정의하기 위해 프로덕트 라인을 구성할 제품의 리스트를 명시하고 제품간의 공통점과 차이점을 식별한다[5]. B2C 쇼핑몰 시스템의 비즈니스 모델에 따른 공통점 및 차이점은 표2와 같다.

표2. B2C 쇼핑몰 시스템 비즈니스 모델에 대한 공통점 및 차이점

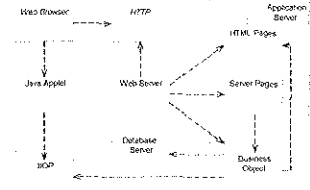
공통점	- 웹 상에서 물건 판매 - *Transaction Processing* - Security - Online Payment - Information Storage - 웹 상에서 물건의 카탈로그 나열 - 주문 정보 접수 - 상품 운송 - 고객 정보 관리
차이점	- 가격이 고정적인지의 여부 - 고객이 원하는 상품의 가격을 결정할 수 있는지의 여부 - 다양한 사이트의 검색 기능을 통해 원하는 가격이나 서비스를 선택할 수 있는지의 여부 - 하나의 상품에 대해 공동구매 가능 여부 - 고객이 판매되는 상품을 자신이 보유한 물품과 교환하거나 금전으로 지불할 수 있는지의 여부 - 향상 상품의 가격을 할인하는지의 여부

3.3 Core Assets

Product Line Scope를 정의한 후에는 프로덕트 라인에 걸쳐 재사용 가능한 것들을 Core Asset으로 정의한다[5]. Core Asset 중 가장 중요한 것은 아키텍처이다. 아키텍처는 프로그램 또는 시스템 컴포넌트들의 구조 및 상호관계, 그리고 설계와 개발에 적용하는 원칙 및 적용지침을 의미한다[7]. B2C 쇼핑몰 시스템 개발을 위한 프로덕트 라인 아키텍처의 가장 큰 특징은 변화에 대한 탄력성이다[5]. 프로덕트 라인 자체가 생산 공정을 나타내기 때문에, 프로젝트 계획 단계에서 도출된 공통점과 차이점을 바탕으로 변화를 수용할 수 있는 아키텍처를 설계하는 것이 가장 중요하다.

Product Line Scope 중 제품간의 공통점으로 추출된 사항은 아키텍처에 Core Asset으로 포함되고, 차이점으로 추출된 사항은 변경 가능하거나 대체 가능한 컴포넌트로 포함된다[5]. <그림 4>는 B2C 쇼핑몰 시스템의 기본 아키텍처를 보여준다.

웹 브라우저는 고객과 쇼핑몰 점원이 사용하는 일반적인 사용자 인터페이스 역할을 수행한다. HTTP는 인터넷 상에서 클라이언트와 서버 사이의 가장 일반적인 연결로서, 아키텍처 상에서 웹 브라우저와 웹 서버 사이의 연결 특성이 비지속성 연결(Connectionless)이라는 것을 표현한다. HTML 페이지와 서버 페이지는 HTTP 연결을 통해 전달된다. 웹 서버는 웹 브라우저가 쇼핑몰 시스템에 연결하기 위한 연결점 역할을 수행하며, 고객 또는 점원의 요구에 따라 HTML 또는 서버 페이지를 선달한다. 웹 브라우저의 요청이 서버 측의 특정 프로세싱을 필요로 하는 경우, 예를 들어 특정 상품에 대한 검색을 웹 브라우저에서 요구하였다면, 웹 서버는 웹 브라우저의 상품검색 요청을 어플리케이션 서버로 전달한다. 어플리케이션 서버는 상품을 검색하는 기능을 보유한 비즈니스 객체를 호출, 구동시킨다. 호출된 비즈니스 객체는 데이터베이스 서버에게 웹 브라우저가 요구한 상품에 대한 검색질의를 전달하고, 데이터베이스 서버는 이를 처리하여 검색된 결과를 질의를 전달한 비즈니스 객체로 전달한다. 상품에 대한 검색 결과는 비즈니스 객체에 의해 검색결과 서버 페이지로 구성되어 웹 서버로 전달되고, 웹 서버는 이를 요청한 웹 브라우저로 전송한다. 만일 웹 브라우저 측에서 그래프와 같은 동적인 요소의 표시가 필요할 경우 자바 애플릿이 사용될 수 있다. 애플릿은 웹 서버 측에서 웹 브라우저로 필요에 따라 전송된다. 웹 브라우저로 전송된 애플릿은 JRMP(Java Remote Method Protocol) 또는 CORBA IIOP를 사용하여 서버 측의 다른 객체들과 통신할 수 있다. 어플리케이션 서버는 비즈니스 객체의 실행을 담당하며, 웹 서버와 같은 머신에서 구동할 수 있다.



<그림 4> B2C 시스템의 기본 아키텍처

아키텍처 이외의 중요한 Core Asset은 표3과 같다[5].

표3. 아키텍처를 제외한 중요 Core Asset

Core Asset	Description
Test Plan, Test Case	각각의 유닛과 시스템 통합에 대한 테스트 계획을 작성
All Design Document	Core Asset Development 프로세스 수행 중에 산출된 모든 디자인 문서
Software Component for Systematic Reuse	프로덕트 라인에 걸쳐 재사용될 Software Component 목록
Product Line Requirement Specification	B2C 쇼핑몰 시스템 프로덕트 라인에서 개발될 프로덕트의 목록과 특징을 기술
Domain Model	개발하고자 하는 프로덕트의 도메인 모델
COTS component	B2C 시스템 구축을 위해 쉽게 획득할 수 있도록 미리 만들어진 제품

Attached Process	Core Asset이 Product Development 프로세스 시에 어떻게 사용될지와 변경될지를 기술
Design/Interface Specification	B2C 시스템을 위한 사용자 인터페이스와 화면 디자인은 HTML 태그로 작성된 Web Page로 구성

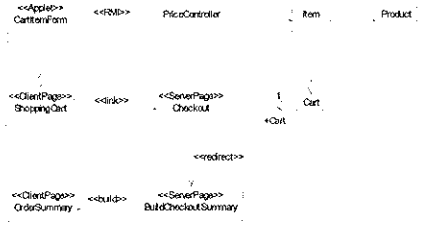
COTS 컴포넌트는 Production Constraints에서 정의한 운영 플랫폼에 적합하고 구입 가능한 상용 컴포넌트를 선정한다. 표4는 Windows NT/2000 Server 플랫폼 상에서 쇼핑몰 시스템을 구축하기 위해 선정한 COTS 컴포넌트의 예를 보여준다.

<그림 5>는 참고문헌 [1]에 소개된 UML의 확장 메커니즘을 적용하여 설계한 클래스 다이어그램으로 B2C 쇼핑몰 시스템의 기능 중 고객이 원하는 상품을 장바구니에 담은 후 서버에 요청을 전송하는 데 필요한 클래스이다.

표4. 선정된 COTS 컴포넌트의 예

소프트웨어 종류	소프트웨어 제품 명
Web Browser	MS Internet Explorer or Netscape Navigator
Payment Gateway	XPAY Gateway 2.0
ORB	Inprise Visibroker for Java
Database Server	Oracle 8i
Web Server	MS IIS 4.0
Application Server	MS MTS 2.0
Multimedia Server	RealSystem
Operating System	Windows NT 4.0/2000 Server

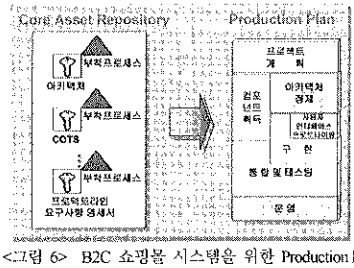
고객이 ShoppingCart 클라이언트 페이지에서 자신이 원하는 상품의 목록을 선택한후 Checkout 서버 페이지로 요청을 전송하면 Checkout 서버 페이지는 이 요청을 BuildCheckoutSummary로 리다이렉트 시킨다. BuildCheckoutSummary 서버 페이지는 결과를 출력하는 OrderSummary 클라이언트 페이지를 생성한다. 클라이언트 페이지에 포함된 ClientForm 애플릿은 서버 측 객체인 PriceController와 IIOP 프로토콜을 통해 통신을 하여 동적인 상품의 정보를 클라이언트 페이지에 반영한다. 이와 같은 디자인에 관련된 모든 다이어그램은 All Design Document에 포함될 수 있다.



<그림 5> B2C 쇼핑몰 시스템의 설계 예

3.4 Production Plan

Production Plan은 각 Core Asset에 대한 부착 프로세스가 어떻게 조화되고 연결될 것인지를 정의하며 프로덕트간의 변화를 제공하는 정확한 배개체를 기술한다[5]. <그림 6>은 B2C 쇼핑몰 시스템 개발을 위한 Production Plan을 구축한 것이다.



<그림 6> B2C 쇼핑몰 시스템을 위한 Production Plan

프로젝트 계획 단계에서는 사용자 요구사항을 획득하고 이를 분석하며, 프로젝트의 크기 및 기간에 대한 예측을 수행한다. 또한 프로젝트 계획에 대한 문서들이 준비된다. 이와 더불어 스케줄의 작성과 자원할당이 포함된다. 아키텍처 정제 단계는 수집, 분석한 사용자 요구사항을 바탕으로 기본 B2C 아키텍처를 개발자의 필요에 따라 정제하는 단계로서, 사용자 요구사항을 반영하고, 구축하는 B2C 쇼핑몰 시스템의 비즈니스 특성에 적합하게 아키텍처를 재정의 한다. 아키텍처의 각 부분은 기존 컴포넌트의 획득 또는 직접적인 개발을 통해 확보되며, 획득된 컴포넌트들에 대해서는 통합 및 테스팅 단계를 수행한다. 예를 들어, 새로이 획득한 컴포넌트는 획득되었거나 개발된 컴포넌트들과 프로세스 초기에 통합 및 테스팅 될 수 있다. 아키텍처 정제 단계 수행 후 고객과 운영자를 위한 사용자 인터페이스 프로토타이핑 단계를 수행한다. 사용자 인터페이스 프로토타이핑의 목적은 사용자 요구사항을 만족시키고, 사용자의 의도를 고려한 인터페이스를 설계하는 데 있다. 컴포넌트 획득 단계에서는 재시한 사용자 요구사항과 아키텍처를 만족하는 소프트웨어 컴포넌트를 최종 선정, 획득하는 작업을 수행한다. 구현 단계는 사용자 요구사항과 아키텍처의 분석을 통해 파악한 구현 대상 컴포넌트에 대해 설계 활동을 수행한다. 구현 대상 컴포넌트에 대한 요구사항은 현재의 구현 부분에 해당하는 사용자 요구사항을 따른다. 통합 및 테스팅 단계는 구현 단계가 완료되기 전에 시작할 수 있다. 통합 및 테스팅은 컴포넌트를 획득하거나 개발한 후 즉시 수행한다. 프로젝트 계획 단계에서는 개략적인 테스팅 전략에 대한 계획이 수립되어야 한다. 테스팅 전략은 컴포넌트 구현의 진행 규정과 컴포넌트 통합 및 테스팅의 진행 규정을 제시한다. 아키텍처 정제 단계를 통해 테스팅 전략은 정제될 수 있다. 컴포넌트 통합을 완료한 후에, 시스템 테스트를 수행한다. 모든 사용자 요구사항이 만족되었는지를 시험하고, 만족도는 어느 정도인지 테스트되어야 한다. B2C 쇼핑몰 시스템의 경우 일반 정보시스템의 유지보수와 달리 지속적인 상품 업데이트와 디자인 변경이 따르고 주기적인 이벤트가 요구된다. 따라서 운영 단계를 통해 B2C 쇼핑몰 시스템 운영을 감시하고 지속적인 컨텐츠 개발과 디자인 개선을 위한 사후 정보를 수집한다.

4. 결론

본 논문의 B2C 쇼핑몰 시스템을 위한 프로덕트 라인은 쇼핑몰에 대한 기본적인 사용자 요구사항을 바탕으로 Product Line Scope, Core Asset, Production Plan을 정의하여 제시하였다. 웹 상에서의 물건 판매, 보안, 온라인 지불, 주문 및 고객 정보를 관리해야 하는 공통점과 가격 정보의 고정 여부, 공동구매, 교환 등의 부가 기능에 따른 차이점을 B2C 쇼핑몰 시스템의 Product Line Scope로 정의할수 있으며 이들은 각각 Core Asset 정의의 기준이 된다. Core Asset은 아키텍처, 테스트 계획, 모든 설계 문서, 재사용 가능한 컴포넌트 목록, 도메인 모델, COTS 컴포넌트, 사용자 인터페이스 등이 있다. 각각의 Core Asset은 시스템 구축 시 어떻게 사용되고 변경되는지가 기술되어 있는 부착 프로세스를 가지고 있으며, 이 부착 프로세스를 연계해 Production Plan을 정의할 수 있다. B2C 쇼핑몰 시스템의 Production Plan은 이미 정의되어 있는 아키텍처를 바탕으로 초기에 구축할 시스템의 사용자 요구사항에 맞게 아키텍처를 정제하는 것과 사용자 인터페이스 프로토타이핑에 중점을 두었다. 쇼핑몰 시스템들은 아키텍처나 기능들이 비슷하고 COTS나 재사용 가능한 컴포넌트를 사용해 구축되어진다. 또한 구축 과정이 어느 정도 일반화되어 있어 구축 과정을 프로덕트 라인화 하면 생산성 향상을 기대할 수 있다. 재시한 B2C 쇼핑몰 시스템을 위한 프로덕트 라인은 사용자 요구사항 분석과 아키텍처 확립에 대한 노력을 감소시키고, 기존 COTS 제품과 이미 검증된 컴포넌트 재사용에 중점을 두었으므로 효율성 및 능률성 측면에서 좋은 성과를 거둘 수 있을 것이다. 또한 이를 통해 구축한 시스템은 이미 검증된 COTS 제품과 컴포넌트를 재사용 함으로 견고하고 안정적이며 신뢰할 수 있을 것이다.

향후에는 프로덕트 라인에 맞추어 많은 수의 쇼핑몰 시스템을 개발해 보고 이에 따른 피드백을 통해 프로덕트 라인을 정제해 가는 연구가 진행되어야 할 것이다. 또한 프로덕트 라인에서 사용할 수 있는 다양한 미들웨어 및 COTS 제품에 대한 로드맵의 준비가 필요하다.

5. 참고 문헌

[1] Jim Conallen : Building Web Applications with UML, Addison Wesley, 1999.
[2] H.M.Deitel, P.J.Deitel, T.R.Nieto : e-Business & e-Commerce HOW TO PROGRAM, Prentice Hall, 2001
[3] Jay M. Tenenbaum, Tripatinder S. Chowdhry, and Kevin Hughes : Eco System : An Internet Commerce Architecture, IEEE Computer, May, 1997, pp.48~55
[4] 심우곤, 백인섭, "컴포넌트의 재사용성 재고를 위한 포괄적 프로세스 모델", 제2회 한국 소프트웨어공학 학술대회 논문집
[5] Carnegie Mellon Software Engineering Institute. A Framework for Software Product Line Practice - Version 3.0
[6] 정기원, 최윤석, "아키텍처 정제에 기반한 쇼핑몰 시스템 개발 모델", 한국프로젝트관리기술회, 프로젝트 관리 기술 논문집, 제1권, 제1호, pp. 5-14, 2000. 10
[7] Len Bass, Paul Clements, and Rick Kazman: Software Architecture in Practice, Addison-wesley, 1998
[8] Paul Clements, "Software Architecture", 제3회 한국 소프트웨어 공학 학술대회 논문집