프로덕트 라인 기반의 웹 학습 시스템 개발

김 행 곤[†] · 김 수 연[††]

요 약

다양하고 빠르게 변화하는 사용자의 요구사항을 만족시키기 위하여 재사용을 고려한 애플리케이션의 개발이 요구되어지고 있다. 이를 통해 대두된 것이 프로덕트 라인이다. 프로덕트 라인은 다양하게 빠르게 변화하는 시장의 요구사항과 특정 도메인 영역에 속하는 애플리케이션 간의 재사용 가능한 아키텍처 및 컴포넌트의 구성으로부터 연관된 시스템 구축 시 생산성과 품질의 향상을 제공함으로써 현재 많은 관심의 초점이 되고 있다. 또한, 이들 프로덕트 라인에서는 프로덕트들 사이의 공통성과 변화성에 초점을 두고 이들 분류 방법으로 휘처 모델링 이라는 개념을 주로 사용하여 분석하고 있다. 또한 재사용 가능한 아키텍처는 많은 변화 계획들과 메커니즘을 포함하고 있다. 그러나 지금까지 이러한 변화들이 일어나는 상황을 이해하는 것과 특별한 상황에서도 가능하게 하는 옵션들을 기록하는 것은 명확히 이루어지지 못하였다. 아키텍처가 오랜 기간 동안 많은 프로덕트 버전에서 사용되어 진다거나, 다른 프로덕트들의 설계를 위해 사용되어지는 아키텍처에서의 프로덕트 라인 문맥에서라면, 매우 중요하게 다루어진다. 즉, 명백한 변화성의 표현과, 아키텍처에서 변경이 되는 적절한 위치를 식별하는 것이 중요하다. 하지만, 아키텍처를 설계하기 위한 아키텍처에서의 변화성 관리에 대한 명확한 방법이 미흡하다. 따라서 본 논문에서는 재사용 가능한 아키텍처를 설계하기 위해 변화성의 명확한 표현과 아키텍처에서의 적절한 위치를 식별하기 위해, 다양한 변화성 타입을 정의하고, 휘처 모델을 기반으로 한 아키텍처의 변화성과 아키텍처의 컴포넌트 관련성에서의 변화성 표현 방법을 기술하고, 제시한 이론을 기반으로 웹 학습 시스템의 설계과정을 거쳐 구현하고자 한다.

키워드 : 프로덕트 라인, 웹기반 학습 시스템, 휘처 모델

Web Learning Systems Development based on Product Line

Haeng-Kon Kim[†] · Su-Youn Kim[††]

ABSTRACT

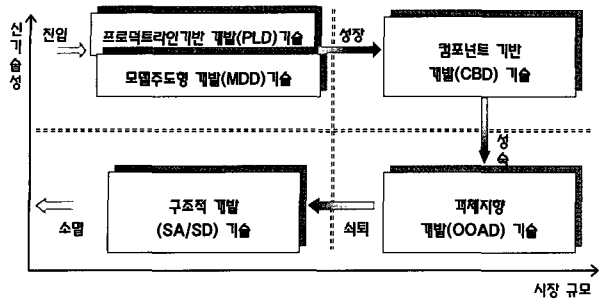
Application developers need effective reuseable methodology to meet rapidly changes and variety of users requirements. Product Line and CBD(Component Based Development) offer the great benefits on quality and productivity for developing the software that is mainly associate with reusable architectures and components in a specific domain and rapidly changing environments. Product line can dynamically focus on the commonality and variety feature model among the products. The product line uses the feature modeling for discovering, analyzing, and mediating interactions between products. Reusable architectures include many variety plans and mechanisms. In case of those architecture are use in product version for a long time, It is very important in architecture product line context for product line design phase. Application developer need to identify the proper location of architecture changing for variety expression. It is lack of specific variety managements to design the product line architecture until nowdays. In this paper, we define various variety types to identify the proper location of architecture changing for variety expression and to design the reusable architecture. We also propose architecture variety on feature model and describe variety expression on component relations. We implemented the web learning system based on the methodology. We finally describe how these methodology may assist in increasing the efficiency, reusability, productivity and quality to develop an application. In the future, we are going to apply the methodology into various domain and suggest international and domestic's standardization.

Key Words : Product Line, Web Based Instruction System, Feature Model

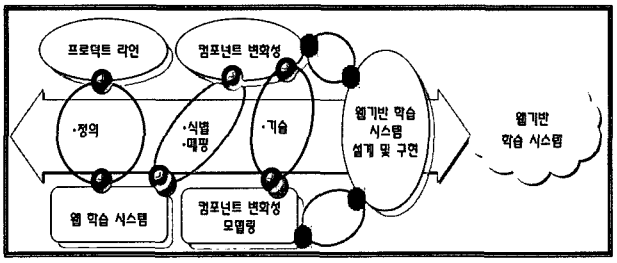
1. 서 론

최근 다양하고 빠르게 변화하는 사용자의 요구사항을 만족시키기 위하여 재사용을 고려한 애플리케이션의 개발이 요구되어지고 있다. 이를 통해 대두된 것이 프로덕트 라인이다. 소프트웨어 프로덕트 라인은 공통의 유사한 기능을 가지고 있는 소프트웨어 시스템 집합으로서, 특정영역의 시장과 용도의 요구사항을 만족하고 미리 구축된 소프트웨어 핵심자산으로부터 정해진 방식으로 개발되어지는 생산기술이다. 다양하게 변화하는 사용자 요구사항과 시스템 개발을 위한 특정 도메인 영역에서 프로덕트 재사용에 관한 연구로

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음
† 종신회원 : 대구가톨릭대학교 컴퓨터공학과 교수
†† 준 회 원 : 대구가톨릭대학교 전자계산교육학과 석사과정
논문접수 : 2005년 2월 23일, 심사완료 : 2005년 5월 23일



(그림 1) 소프트웨어 생산 기술 발전 추세



(그림 2) 연구의 핵심 기술

프로덕트들 사이의 공통성과 변화성에 초점을 두고 체계적인 접근을 제공한다. 또한, 재사용 가능한 핵심자산을 기반으로 조립·생산함으로써 생산성 향상과 품질확보 등의 많은 장점을 얻을 수 있다[1, 2]. (그림 1)은 소프트웨어 생산 기술의 발전 추세를 나타내고 있다. 구조적 분석/설계 개발 기술은 이미 소멸 되어진 상태이고, 객체지향 분석 개발 기술 역시 이미 쇠퇴되어지는 상태이다.

컴포넌트 기반 개발 기술은 많은 연구를 통하여 이미 성숙되어졌고, 시장 규모면에서 큰 부분을 차지하고 있고, 기술성도 높다고 볼 수 있다.

이에 컴포넌트 기반 개발 기술을 포함하고 있는 프로덕트 라인 기반 개발기술과 모델주도형 개발 기술이 새로이 등장하고 있다. 프로덕트 라인 기반 개발 기술의 목표는 일련의 유사한 소프트웨어 시스템의 공통성과 구별되는 특성을 이해하고 제어함으로써 시스템의 체계적인 개발을 지원하는 것이다[3, 4, 5].

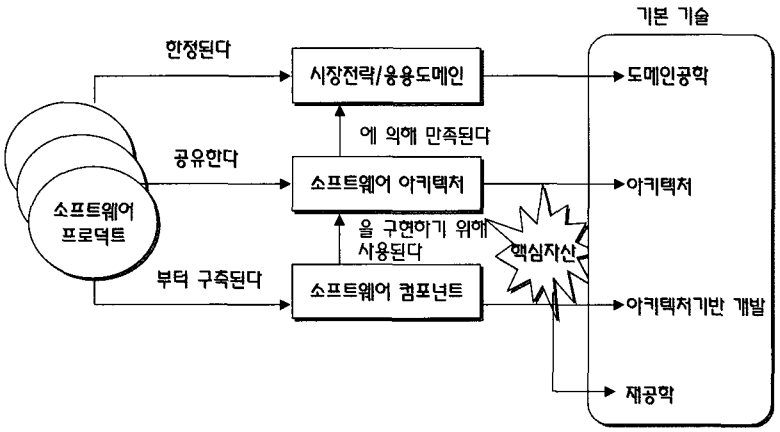
컴포넌트 가변성을 이용하여 사용자가 용도에 알맞게 기능을 특화할 수 있고 다양하고 빠르게 변화하는 시장의 요구사항과 특정 도메인 영역에 속하는 애플리케이션간의 재사용 가능한 아키텍처 및 컴포넌트의 구성으로부터 연관된 시스템 구축 시 생산성과 품질의 향상을 제공함으로써 현재 많은 관심의 초점이 되고 있다. 본 논문에서는 프로덕트 라인을 기반해서 휘처 모델링을 통한 아키텍처 기술에서의 다양한 타입의 변화성을 기술하고, 변화성의 타입을 표기하는 방법과 변경이 지원되는 아키텍처에서 위치를 쉽게 찾기 위해 컴포넌트 변화성에 대해 제시하고, 제시한 이론을 기반으로 웹기반 학습 시스템의 설계과정을 거쳐 구현하고자 한다.

2. 관련 연구

2.1 프로덕트 라인 접근

소프트웨어 프로덕트 라인은 공통성을 공유하는 소프트웨어 집약적인 시스템 집합으로 특정 시장 부문의 필요성을 만족하고 공통 핵심자산 집합으로부터 개발되어진 특징들의 집합이다. 즉 프로덕트는 시장 전략과 애플리케이션 도메인에 적합하고, 아키텍처를 공유하고 기존 컴포넌트로부터 구축되어진다. 프로덕트 라인 아키텍처는 연관된 프로덕트 집합과 조직에 의해 개발된 시스템을 위한 공통 아키텍처로 향상된 생산성과 소프트웨어 품질을 제공한다[3, 4].

프로덕트 라인 기반 개발은 기술적, 구조적으로 정의된 관리에서 핵심자산 개발과 프로덕트 개발을 포함한다. 핵심자산 개발과 프로덕트 개발에서 새로운 프로덕트들은 핵심자산으로부터 만들어지고 핵심자산들은 기존 프로덕트로부터 획득된다. 프로덕트 라인과 관련된 연구는 아키텍처 정의와 전개, 컴포넌트 개발, COTS 활용, 기존 자산 마이닝과 소프트웨어 시스템 통합 등이 있다. 이들은 실제 핵심자산과 프로덕트를 생성하고 전개하는 적절한 기술 적용에 대한 필요성이다. 프로덕트 라인은 (그림 3)과 같이 특정 시스템



(그림 3) 프로덕트 라인

시장을 목표하고 있는 시스템이며 이 시스템 패밀리를 위한 기반이 될 수 있고 재사용 가능한 모델이며 공통 자산을 통해 만들어진다. 컴포넌트가 플러그 인 될 수 있는 프레임워크를 제공하는 아키텍처를 기반으로 필요한 컴포넌트를 선택적으로 조립함으로써 시장 요구에 맞는 시스템을 생산할 수 있다[5, 11].

2.2 휘처 모델

휘처 모델은 프로덕트 라인 요구사항 분석의 핵심 결과로 프로덕트 라인 구성의 기능적, 비기능적인 요소 뿐 아니라 공통성과 변화성을 획득하고 다양한 관점을 제공할 수 있다. 사용자는 휘처 모델을 통해 프로덕트 라인의 기능을 이해할 수 있고, 개발자들은 다양한 프로덕트 변화의 개발을 유도할 수 있다[6, 12].

(그림 4)는 휘처 모델의 간략한 예를 보여 주고 있다. 분석 모델로서의 휘처 모델링은 하나의 모델로 휘처들의 의미를 표준화하고, 통합함으로써 표준 용어들을 정의하고 문제들에 대한 손쉬운 의사소통을 가능하게 하며 응용 시스템들 사이의 공통점과 차이점을 평가하기 위한 잣대로 사용될 수 있으며, 한 영역을 특성화하고 다른 영역들과 비교하기 위하여 사용될 수 있다.

즉, 휘처 모델링은 영역 공학의 한 방법으로서 영역 언어를 중심으로 영역 내의 공통점과 차이점을 찾아내어 구조화하여 분석하고, 분석된 결과를 바탕으로 영역에 적합한 설계 모델을 만들고, 영역 내에서 가치있는 소프트웨어 컴포넌트를 추출해 낸다. 휘처 모델링을 기반하여 추출해 낸 재사용성 있는 소프트웨어는 시스템 개발 초기부터 재사용성을 고려하여 생산되었기 때문에, 분석 대상 영역에서 변화 가능한 많은 응용 시스템에 대하여 적응성이 높고, 분석 단계 모델을 통해 재사용 될 수 있다는 점에서 매우 가치가 높다고 할 수 있다[7, 8]. 앞서 기술한 바와 같이 휘처 모델은 영역내의 용어를 중심으로 하여 영역을 분석하기 때문에, 영역 내의 특성을 최대한 살려서 표현할 수 있게 된다. 이러한 영역 용어를 기반으로 추출된 휘처들은 다른 영역과 차별되어 드러나는 것으로서, 한 영역을 특성화하고 비교하는데 도움을 준다. 휘처의 타입은 응용 시스템에서의 휘처 존재 여부에 따라 결정된다. 즉, 영역 내의 모든 응용 시스템에 항상 필요한 휘처인가, 있어도 되고 없어도 되는 휘처인가가 주요 관심사이다. 휘처의 타입은 아래의 <표 1>과 같이 구분되어진다.

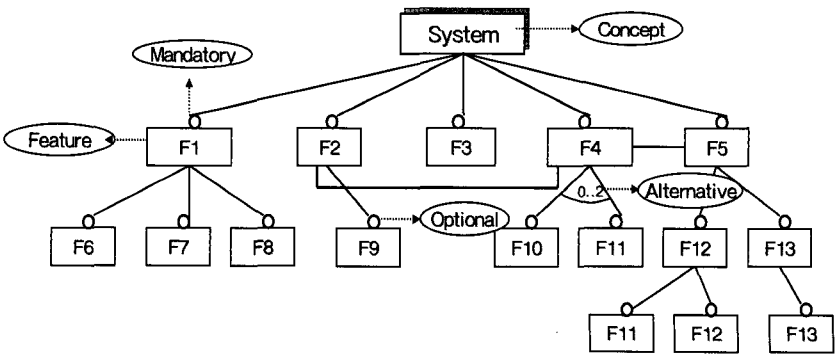
2.3 웹기반 학습 시스템

기존의 컴퓨터 활용 수업은 컴퓨터 보조 수업(CAI)의 형태로 이루어져왔으나, 최근 웹의 대중화로 인하여 기존의 CAI와 웹을 통합한 웹기반 수업(WBI)에 대한 관심이 높아지고 있다. 웹기반 수업이란 특정하게 미리 계획된 방법으로써 학습자의 지식이나 능력을 육성하기 위한 의도적인 상호작용을 웹을 통해 전달하는 활동을 말하며, 학습을 촉진하고 지원하는 의미 있는 학습 환경을 만들기 위해 웹의 속성과 자원을 활용하는 하이퍼미디어 기반의 수업 프로그램이다. 웹기반 수업의 발달은 네트워크 통신 기술의 발전과 그것의 교육적 활용에 바탕을 두고 있다고 볼 수 있으며, 네트워크가 교육에 활용된 형태는 크게 세 가지 정도로 구분되어 진다. 첫째, 네트워크를 면대면 교육이나 원격교육 등에서 하나의 보조적 매체로 활용하는 형태가 있다. 둘째, 네트워크 자체가 하나의 수업으로 활용되는 형태가 있는데 이것은 설계가 잘된 웹기반 수업이라면 어떠한 주제에서건 학습자 주도적(self-directed)이고 학습자의 속도(self-paced)에 맞는 교수 방법을 제공하며, 다양한 매체 중심의 환경으로 구성되어 있기 때문에 학습자에게 웹 브라우저의 활용과 인터넷 접속을 촉진시키는 역할을 한다.

셋째, 네트워크를 보다 자유로운 지식 네트워킹의 장, 토론의 장, 토론의 참여수단, 온라인 데이터베이스 활용의 수단 또는 세계에 흩어진 전문가나 동료들과의 정보교환의 수단 등으로 이용하는 형태이다.

〈표 1〉 휘처의 타입

휘처 타입	설 명
필수적 (Mandatory)	모든 응용 시스템에 반드시 있어야 하는 휘처
선택적 (Optional)	응용 시스템에 따라 있을 수도 있고 없을 수도 있는 휘처
택일적 (Alternative)	선택적으로 추가되어질 수 있는 휘처중의 하나가 추가되는 경우의 휘처



(그림 4) 다이어그램으로 표현된 휘처 모델

3. 프로덕트 라인 아키텍처에서의 컴포넌트 변화성

프로덕트 라인 아키텍처에서의 컴포넌트 변화성에 대한 내용으로, 동일한 프로덕트 군에 속하는 프로덕트들은 많은 공통성을 가지지만 또한 프로덕트 사이의 변화성도 있다. 변화성은 다양한 사용자와 다양한 설계, 그리고 구현 요구사항들에 따라 제공되고, 도메인 분석 시에 식별된다. 따라서 변화성은 프로덕트 라인 아키텍처 설계 단계에서 고려되어지고, 코드 레벨이 아닌 아키텍처 레벨에서 다루어져야 한다. 또한 변화성은 컴포넌트 조립 시에 컴포넌트 행위가 변경될 수 있는 특정 변화점에서 가능하고 컴포넌트 설계 동안에 결정된다. 따라서 제 3 장에서는 아키텍처 상에서의 휘처 모델을 제시하고, 이 때의 컴포넌트들 사이의 변화성을 설계한다.

3.1 아키텍처상의 휘처 모델

휘처 모델은 다양한 변화성을 보여주고, 요구사항 사이의 변화성을 만족하기 위해 아키텍처에서의 변화점을 제시하고 어떻게 변화점을 표현할 것인지의 가이드라인을 제공한다. 또한 개발자 측면에서는 다양한 프로덕트 변화의 개발을 유도하고, 사용자 측면에서는 프로덕트 라인의 기능을 쉽게 이해할 수 있도록 도와주는 기능을 제공하기도 한다. (그림 5)에서 제시된 점선 부분을 통해서 아키텍처 상에서의 변화성을 설계해 보고자 한다.

3.2 컴포넌트의 변화성 타입

아키텍처에서 다양한 요구사항을 표현하려는 것은 프로덕트 라인 아키텍처에서 프로덕트들 사이의 변화성을 다루기 위해 다양한 선택 사항을 포함하는 경우이다. 이러한 변화성의 타입은 <표 2>와 같이 크게 3가지 형태로 정의된다.

변화성의 표기는 <표 3>과 같이 UML의 다중성 표기를 기반으로 5가지 형태로 식별될 수 있다. 문맥에서의 보다 쉬운 이해를 위해 Variant 타입의 기본 표기를 정의한다.

3.3 컴포넌트의 변화성 표현

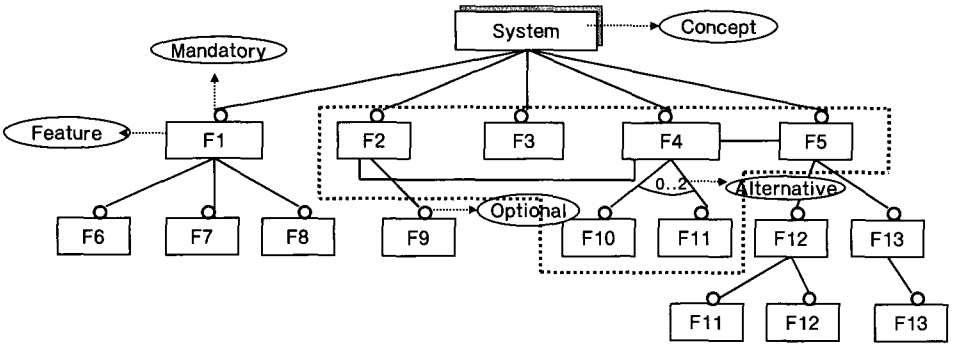
하나의 휘처는 하나의 컴포넌트에 대응된다고 가정했을 때 다음과 같이 변화성을 표현해 볼 수 있다. (그림 6)은 두 프로덕트의 컴포넌트 뷰를 보여준다. 이들 컴포넌트 뷰는 (그림 5)의 점선 부분에서의 어떤 휘처를 선택하느냐에 따라 서로 다른 아키텍처의 표현을 나타낸다. 그림에서 두 프로덕트의 차이점은 컴포넌트 B와 컴포넌트 D의 위치이다. 이들 변화성을 지원하기 위해 아키텍처는 하나의 다이어그램으로 통합될 수 있다. (그림 7)은 하나의 표현으로 두 개의 아키텍처를 표현한다. Variant A_1은 변화점(VP)을 의미하고, 컴포넌트와 구별하기 위해 1차원의 그레이 박스를 사용한다. Variant A_1의 가능한 구현 기능인 컴포넌트 B와 컴포넌트 D를 추적하기 위해 (그림 8)과 같이 나타낼 수 있다.

〈표 2〉 변화성 타입

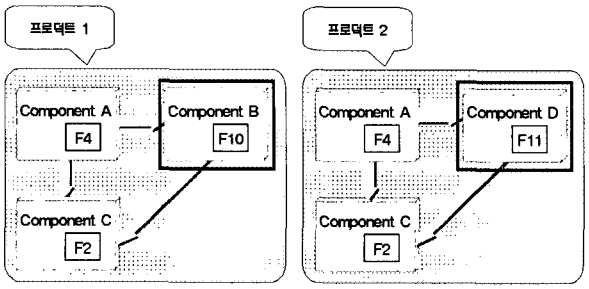
변화성 타입	설 명
Optional	· 어떤 기능이 한 프로덕트에 포함되고 다른 프로덕트에는 불포함 · Optional의 관심은 선택적인 가능성을 갖는 다른 기능들의 관계와 프로덕트에서 포함되어지지 않은 선택적인 기능인 경우 이러한 관계들이 발생할 수 있음을 나타냄
Alternative	· 추가될 수 있는 여러 선택 중의 하나의 추가
Set of Alternative	· 추가될 수 있는 여러 선택 중의 여러 개가 추가

〈표 3〉 변화성 타입 표기의 예

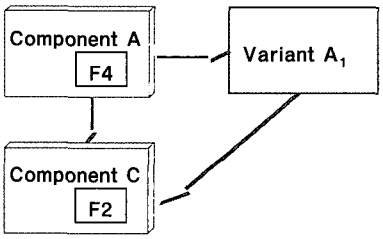
변화성 표기	변화성 타입	의 미
Variant A_0	Optional	프로덕트에 포함되는 정확히 하나의 구현에 존재하는 것
Variant B_1	Alternative	Viriant의 다중 실현이 존재하고, 정확히 하나가 프로덕트에 포함되어야 함
Variant $C_{1..*}$	Set of Alternative	Variant의 다중 실현이 존재하고 그 중의 하나가 프로덕트에 포함될 수 있음
Variant $D_{0..1}$	Optional Alternative	Variant의 다중 실현이 존재하고 그 중의 0 혹은 하나가 프로덕트에 포함될 수 있음
Variant $E_{0..*}$	Optional set of Alternative	Variant의 다중 실현이 존재하고 그 것의 모음이 프로덕트에 포함될 수 있음



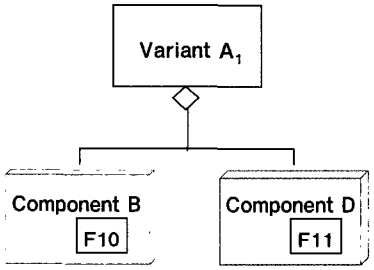
(그림 5) 아키텍처상의 휘처 모델



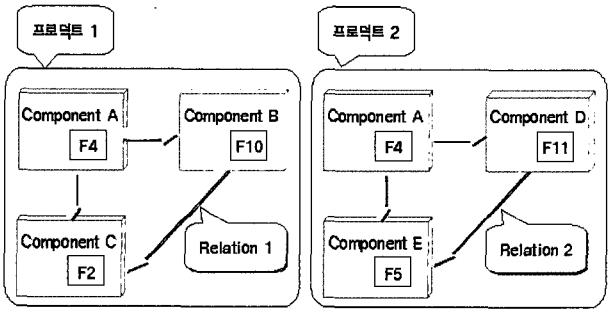
(그림 6) 두개 프로덕트를 위한 선택적인 아키텍처



(그림 7) 아키텍처에서의 변화성



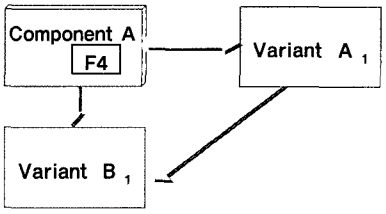
(그림 8) 변화성의 선택



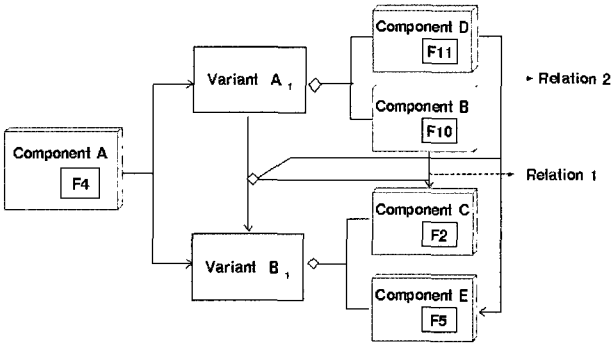
(그림 9) 관련성의 변화성

변화성은 아키텍처의 컴포넌트 사이에서 관련성을 위하여 발생될 수 있다. (그림 9)는 두 프로덕트에 대한 컴포넌트 뷰로 컴포넌트 사이의 관련성에 대한 변화성을 표현한다. 프로덕트 1은 컴포넌트 B와 컴포넌트 C사이에서 Relation 1을 발생시키고, 프로덕트 2에서의 컴포넌트 D와 컴포넌트 E 사이에서 Relation 2를 발생시킨다.

또한, 이들 관련성에 대한 변화는 연결된 컴포넌트의 변화성을 포함한다. 그러므로 (그림 10)에서와 같이 컴포넌트 C와 컴포넌트 E는 Variant B_1로 표현 가능하다. (그림 11)



(그림 10) 아키텍처에서의 변화성



(그림 11) 변화점의 표현

은 Variant A_1과 Variant B_1 사이의 의존성을 하나의 큰 그림으로 통합한 것이다.

4. 프로덕트 라인을 적용한 웹기반 학습 시스템 설계 및 구현

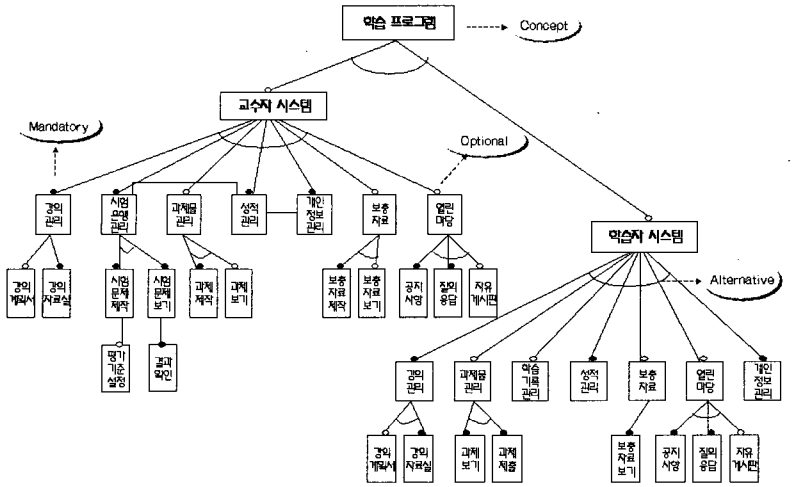
4.1 웹기반 학습 시스템에서의 휘처 모델링

프로덕트 라인의 공통점과 차이점을 분석하기 위한 가장 기본이 되는 휘처 모델은 휘처들 각각의 의미와 성격, 상호 연관성을 잘 나타내고 이를 하나의 모델로 집약한 것이다.

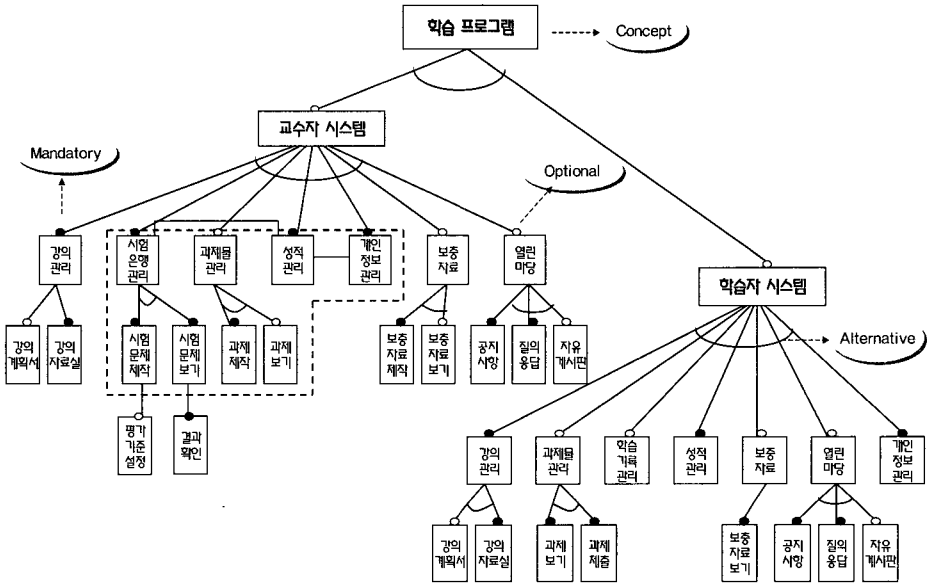
(그림 12)는 웹기반 학습 시스템을 위한 휘처 모델로서, 시스템의 전체적인 의미를 담고 있는 도메인 영역인 학습 프로그램 안에서 교수자 시스템과 학습자 시스템으로 나누어질 수 있다. 강의관리, 시험은행관리, 성적관리, 개인정보관리 등의 휘처는 교수자 시스템에서 필수적으로 포함되어야 하는 요소이고, 공통적으로 사용되는 요소이므로 Mandatory 휘처로 표현되고 있다. 또한 두 개의 시스템 아래 각각의 서브 휘처들을 갖고 있으며 Optional 휘처와 Alternative 휘처의 사용으로 시스템의 변화성을 설명할 수 있게 된다. 아래의 <표 4>는 학습 시스템이 제공하는 교수자 및 학습자 시스템의 상세 기능들을 설명한 것이다.

4.2 웹기반 학습 시스템에서의 변화성 표현

(그림 12)의 휘처 모델링에서 점선 부분의 휘처를 통해 변화성을 제시한다. (그림 13)에서 표현된 휘처들을 살펴보면 시험은행관리, 성적관리, 개인정보관리 등의 휘처는 교수자 시스템에 반드시 포함되어야 하는 Mandatory 휘처로 표현을 하였고, 과제물관리, 보충자료, 열린마당 등의 휘처는



(그림 12) 웹기반 학습 시스템의 휘처 모델



(그림 13) 웹기반 학습 시스템의 휘처 모델에서의 변화성 표현

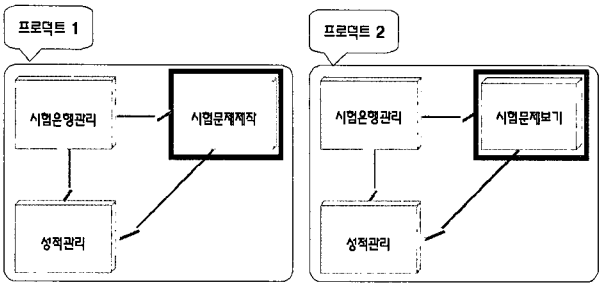
〈표 4〉 휘처의 기능 명세

휘 처	설 명
강의관리	교수자는 강의 관리 휘처를 통하여, 강의 계획서를 업로드 하고 학습자는 강의자료실을 통하여 강의 관련 자료의 획득이 가능함
시험은행관리	교수자는 학습자의 성적처리를 위한 시험문제 제작 및 관리
과제물관리	교수자는 과제를 제작하고 볼 수 있는 기능을 가지며, 학습자는 과제를 보고 제출하는 기능을 갖게 됨
성적관리	교수자는 과제를 제작하고 볼 수 있는 기능을 가지며, 학습자는 과제를 보고 제출하는 기능을 갖게 됨
개인정보관리	수행평가 점수와 중간고사, 기말고사 점수를 누적하여 성적표를 손쉽게 작성
보충자료	학교생활기록부 등을 포함하고 있으며 학교생활기록부는 교과별 성취수준평가, 교과활동, 특별활동, 봉사활동, 자격증 취득, 수상 경력 등의 학생의 학교생활 전반에 대한 내용
열린마당	공지사항과 질의응답 및 자유게시판이 포함됨

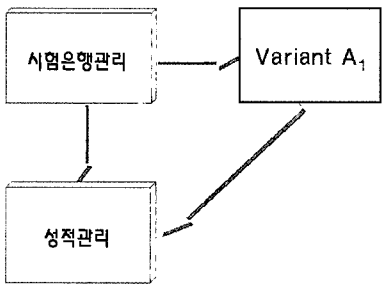
교수자 시스템에 선택적으로 포함될 수 있는 기능으로써 Optional 휘처로 표현되어지고 있다.

(그림 14)는 두 프로덕트의 컴포넌트 뷰를 보여준다. 이들 컴포넌트 뷰는 (그림 13)의 점선 부분에서 어떤 휘처를 선택하느냐에 따라 서로 다른 아키텍처의 표현을 나타낸다. 그리고 휘처는 하나의 컴포넌트에 대응된다고 가정한다. (그림 14)에서 두 프로덕트의 차이점은 시험문제제작 컴포넌트와 시험문제보기 컴포넌트의 위치이다.

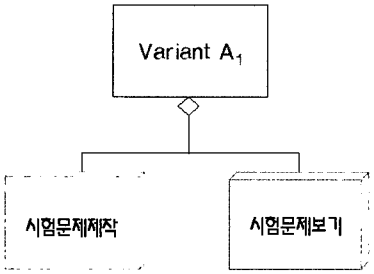
이들 변화성을 지원하기 위해 아키텍처는 하나의 다이어그램으로 통합될 수 있다. (그림 15)는 하나의 표현으로 두 개의 아키텍처를 표현한다. Variant A_1은 변화점(VP)을 의미하고, 컴포넌트와 구별하기 위해 1차원의 그레이 박스를 사용한다. 이들 Variant A의 가능한 구현 기능을 추적하기 위해 (그림 16)과 같이 나타낼 수 있다. 변화성은 아키텍처의 컴포넌트 사이에서 관련성을 위해 발생할 수 있다.



(그림 14) 웹기반 학습 시스템의 두 개 프로덕트를 위한 선택적 아키텍처

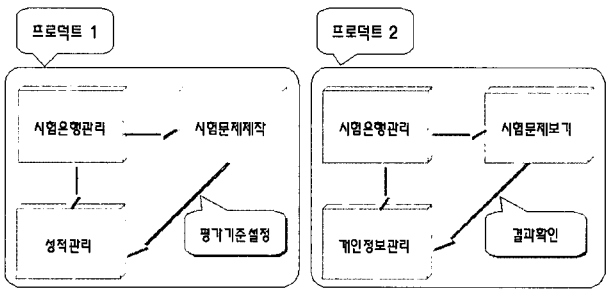


(그림 15) 아키텍처에서의 변화성

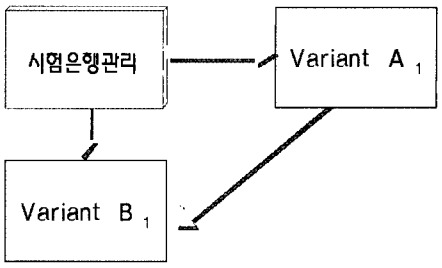


(그림 16) 변화성의 선택

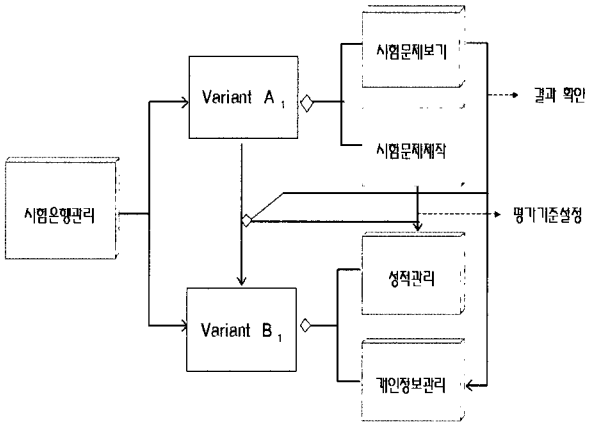
(그림 17)은 두 프로덕트에 대한 컴포넌트 뷰로 컴포넌트 사이의 관련성에 대한 변화성을 표현한다. 프로덕트 1은 시험문제제작 컴포넌트와 성적관리 컴포넌트 사이에서 평가기준설정이라는 관련성을 발생시키고, 프로덕트 2에서의 시험문제보기 컴포넌트와 개인정보관리 컴포넌트 사이에서 결과확인이라는 관련성을 발생시킨다. 또한, 이들 관련성에 대한 변화는 연결된 컴포넌트의 변화성을 포함한다.



(그림 17) 관련성의 변화성



(그림 18) 아키텍처에서의 변화성



(그림 19) 변화점의 표현

그러므로 성적관리 컴포넌트와 개인정보관리 컴포넌트는 (그림 18)과 같이 Variant B_1로 표현 가능하다.

(그림 19)는 Variant A_1과 Variant B_1 사이의 의존성을 하나의 큰 그림으로 통합한 것이다.

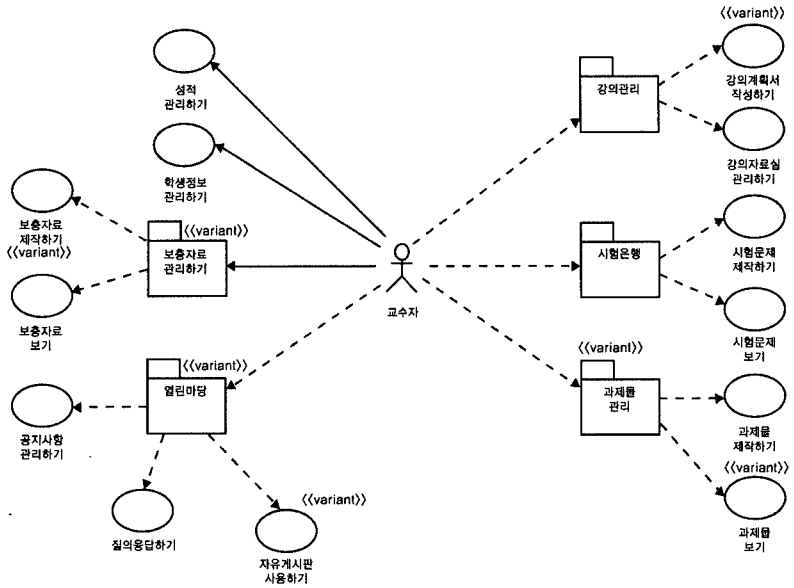
4.3 웹기반 학습 시스템에서의 모델링

본 논문에서는 시스템을 시각화하고, 구축하는데 필요한 산출물들을 문서화하기 위한 모델링 언어로 UML(Unified Modeling Language)을 사용하여 시스템과 사용자의 요구사항을 파악하고 이를 기반으로 사용자와 유스케이스를 추출하고 연관짓는 유스케이스 뷰와, 객체간의 메시지 흐름을 나타내기 위한 인터액션 뷰를 나타내고 이들 정보를 바탕으로 컴포넌트 뷰와 클래스 뷰를 작성한다.

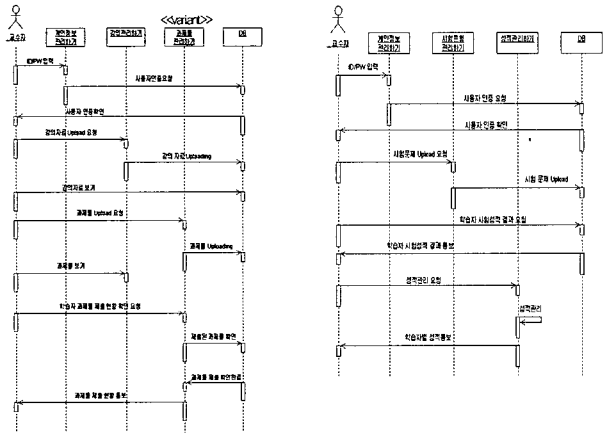
4.3.1 유스케이스 뷰

유스케이스 뷰는 시스템의 유스케이스와 액터, 그리고 이들 간의 관계를 보여준다. 유스케이스는 시스템이 제공하는 기능을 상위 수준에서 바라본 단편들이라고 할 수 있다.

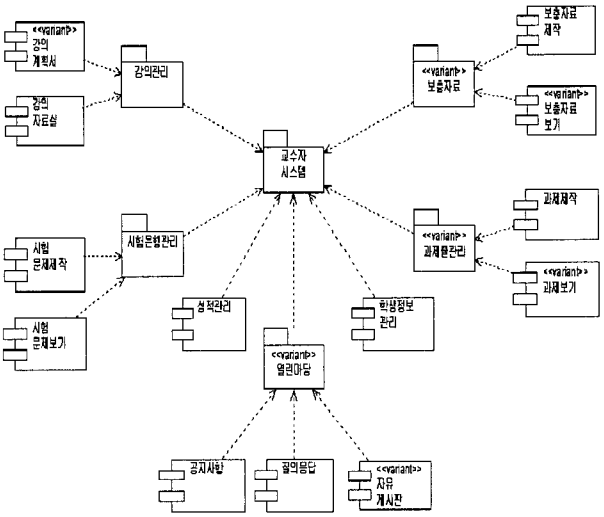
액터는 구축할 시스템과 상호작용하는 사람 또는 사물을 말한다. 유스케이스 뷰에서 시스템 액터와 유스케이스 그리고 이들 간의 관계를 볼 수 있다. (그림 20)은 웹기반 학습 시스템 휘처모델을 바탕으로 유스케이스 뷰를 작성한 것이다. 휘처 모델에서의 변화 가능한 Optional 휘처로 설계된 부분은 <<variant>>라는 스테레오 타입을 사용하여 변화성 있는 부분임을 명시하고 있다.



(그림 20) 교수자 시스템의 유스케이스 뷰



(그림 21) 교수자 시스템의 인터랙션 뷰



(그림 22) 교수자 시스템의 컴포넌트 뷰

4.3.2 인터랙션 뷰

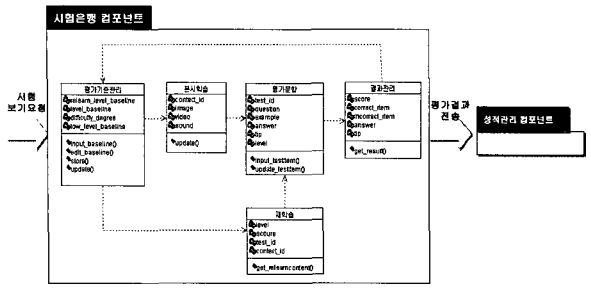
인터랙션 뷰는 해당 유스케이스를 실행하는 업무의 흐름을 단계별로 보여준다. 업무가 흘러갈 때 어떤 객체가 필요하고 서로 간에 어떤 메시지를 주고받는지, 그리고 업무 흐름을 시작하는 액터가 무엇인지, 메시지가 전송되는 순서는 어떻게 되는지 알려준다. (그림 21)은 성적관리 컴포넌트간의 메시지를 진행 순서에 따라 표현한 인터랙션 뷰이다. 왼쪽의 인터랙션 뷰는 교수자 시스템에서 "수업"에 관한 컴포넌트들 사이의 메시지 진행 순서를 나타낸 것이고, 오른쪽 인터랙션 뷰는 교수자 시스템에서의 "평가"에 관한 컴포넌트 간의 메시지 진행 순서를 나타낸 것이다. 역시 Optional 휘처로 설계되어진 "과제물관리하기" 컴포넌트는 <<variant>> 스테레오 타입을 사용하여 변화성을 명시하고 있다.

4.3.3 컴포넌트 뷰

컴포넌트 뷰는 클래스들의 집합으로 물리적 구성단위인 컴포넌트와 그들 간의 구성 및 의존 관계를 표현한 것으로 시스템의 정적인 구현 뷰를 나타낸 것이다. (그림 22)는 웹 기반 학습 시스템의 컴포넌트 뷰를 나타낸 것으로서, 서브 휘처들을 포함한 휘처는 하나의 패키지로 표현하였으며, 시스템에서 변화 가능한 부분으로 사용되어 질 수 있는 휘처들은 대응된 컴포넌트에 <<variant>>라는 스테레오 타입을 사용하여 변화성을 식별하고 있다.

4.3.4 클래스 뷰

클래스 뷰는 시스템에 있는 클래스와 패키지를 표시하기 위해 사용되며, 시스템 내의 여러 부분들과 이들 간의 관계를 정적으로 보여준다. (그림 23)은 앞 단계를 통해 얻은 분석·설계를 이용하여 작성한 교수자 시스템의 평가 컴포넌트에 대한 클래스 모델 뷰이다.

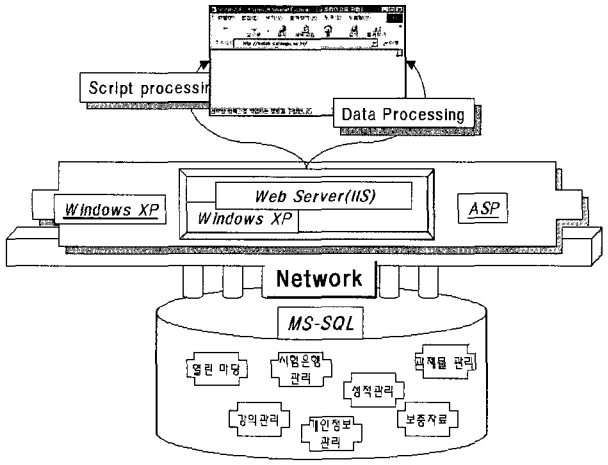


(그림 23) 시험은행 컴포넌트의 클래스 뷰

4.4 웹기반 학습 시스템의 구현

4.4.1 개발 환경

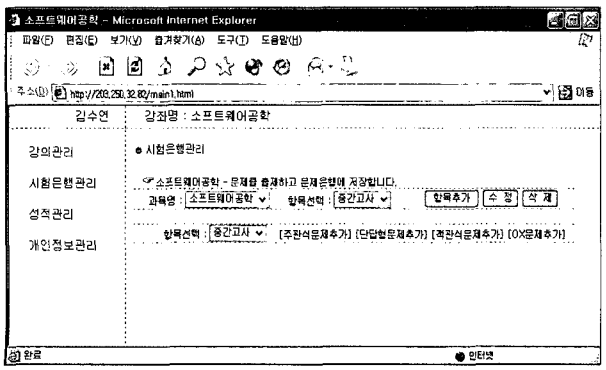
본 논문의 웹기반 학습 시스템을 개발하기 위한 환경은 (그림 24)와 같이 나타낼 수 있다. 개발환경은 서버 시스템으로 windows XP를 사용하고 IIS 상에서 ASP를 통해서 작성되었으며 데이터베이스는 MS-SQL Server를 사용하였다.



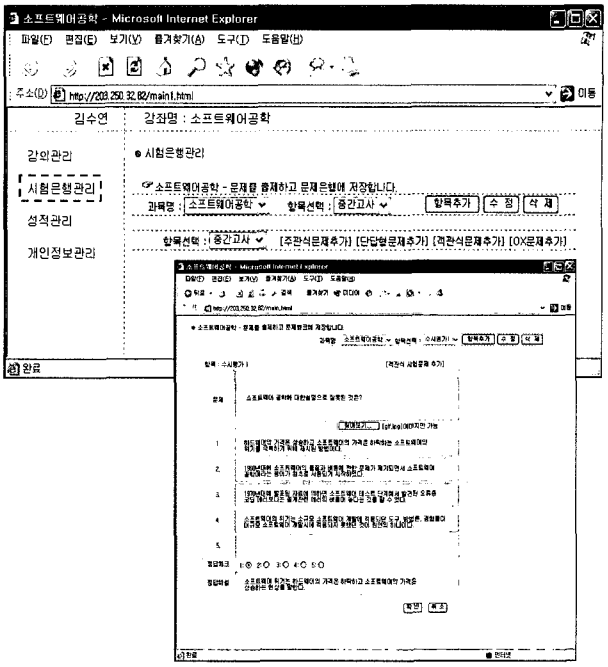
(그림 24) 웹기반 학습 시스템 전체 구현 구조도

4.4.2 웹기반 학습 시스템 실행 및 평가

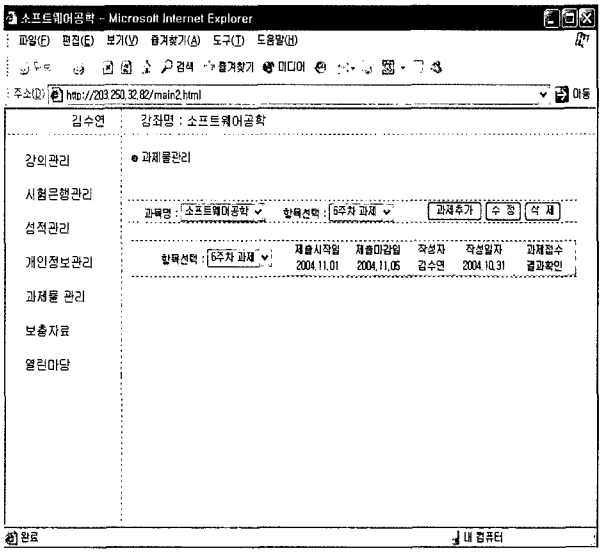
(그림 25)는 교수자 시스템에서 Mandatory 휘처들로 정의된 강의관리, 시험은행관리, 성적관리, 개인정보관리 휘처들로 설계된 웹기반 학습 시스템의 구현 화면을 나타낸 것이다.



(그림 25) Mandatory 휘처로 설계된 학습 시스템 구현 화면



(그림 26) 시험은행관리 컴포넌트의 항목추가 실행화면

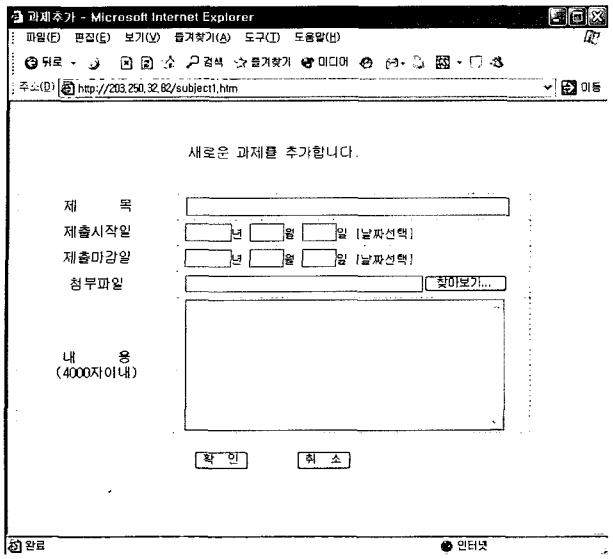


(그림 27) Optional휘처를 포함시킨 웹기반 학습 시스템 구현 화면

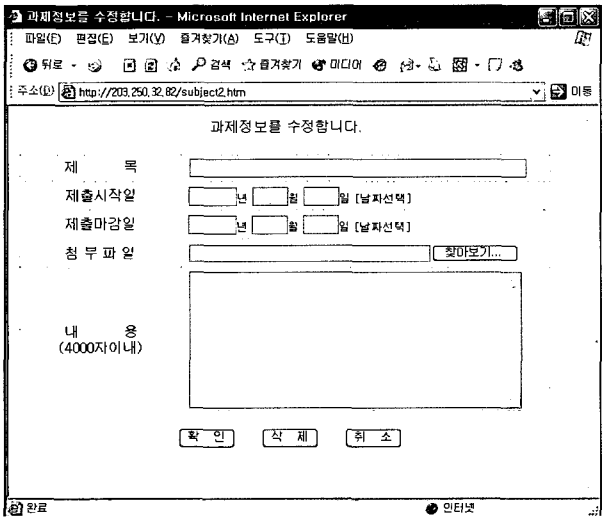
(그림 26)은 웹기반 학습 시스템의 시험은행관리 컴포넌트에서의 항목추가 기능을 실행시킨 화면이다.

(그림 27)은 교수자 시스템에서 Optional 휘처들로 정의된 과제물 관리, 보충자료, 열린마당 휘처들도 모두 포함시켜 설계된 웹기반 학습 시스템의 구현 화면을 나타낸 것이다. 이렇게 프로덕트 라인의 공통성과 변화성의 특성을 활용하여 공통 도메인 내의 유사한 애플리케이션의 개발이 손쉽게 이루어진다.

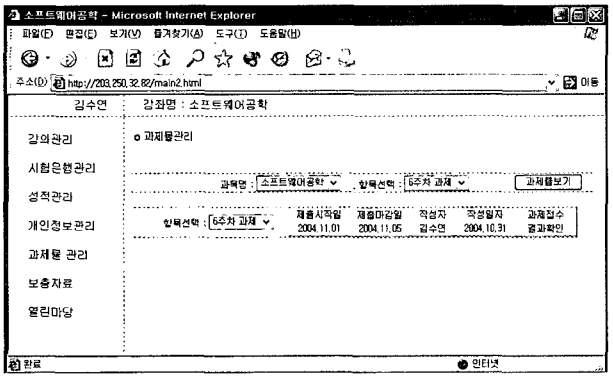
(그림 28)은 과제물 관리 컴포넌트에서 새로운 과제를 추가시키기 위한 실행 화면이다.



(그림 28) 과제물 관리 컴포넌트의 과제추가 실행화면



(그림 29) 과제물 관리 컴포넌트의 추가된 과제 수정화면



(그림 30) 과제물 관리 컴포넌트의 과제물보기 실행화면

(그림 29)는 과제물 관리 컴포넌트에서 추가된 과제를 수정하기 위한 실행 화면이다.

(그림 30)은 과제물 관리 컴포넌트에서 교수자가 과제물 보기 기능을 시스템에 추가시킨 실행화면이다. 과제 추가와 과제물 보기 기능은 설계단계에서 변화 가능한 <<variant>>컴포넌트로 설계된 부분으로 시스템 구현에 있어서의 변화가 가능하다.

4.4.3 웹기반 학습 시스템 평가

<표 5>는 현재 웹상에서 운영, 참조되고 있는 시스템들의 명세 작성 및 검색 특징들을 비교하였다. 명세작성과 검색을 지원하는 유통 사이트가 유사하게 많이 있으나, 분산 환경에서 응용 시스템 개발이 용이하게 하기 위해 분산 컴포넌트를 사용자 측면과 관리 측면, 그리고 저장 측면으로 나누어 명세를 통해 컴포넌트로 만들었으며, 웹학습 시스템 도메인에 맞도록 컴포넌트로의 확장이 가능하다. 또한 사용자도 쉽게 원하는 웹 정보 등록 및 수정을 할 수 있으며, 검색을 통해 원하는 자료를 찾을 수 있다. 이러한 모든 기능은 에이전트를 통해 이루어지므로, 사용자에게 좀 더 빠른 웹 학습 정보 제공과 함께, 자동화된 기능을 통해 많은 노력과 시간을 절약할 수 있으며, 웹 학습시스템 목적인 재사용성을 극대화할 수 있다.

〈표 5〉 웹기반 학습 시스템 평가

항목 시스템	주요기능	개발프로세스 제공여부	명세 제공 여부	검색방법	에이전트 지원여부
Blaze Advisor	규칙 기반 DB 제공 규칙검색, 관리	YES	NO	규칙에 따른 검색	YES
IBM EAI	SCM, ERP, CRM	YES	NO	NO	NO
Rational Rose	UML 형식의 명세	YES	YES	NO	NO
Component Bank	컴포넌트 등록, 검색, 판매	YES	YES	키워드	NO
웹학습 시스템 (본논문)	에이전트 지원의 컴포넌트 정보 등록, 검색	YES	YES	키워드 카테고리	YES

5. 결론 및 향후연구

최근 아키텍처와 재사용 컴포넌트들의 공유를 통한 애플리케이션 개발이 최상의 패러다임으로 인식되고 있다. 컴포넌트를 기반으로 재사용성과 프로덕트 개발의 생산성, 비용 및 품질 향상을 위해 본 논문에서는 효율적인 프로덕트 라인 지원 프로세스와 모델링에 대해 연구하였다. 프로덕트 라인은 그 군 사이에서 아키텍처와 재사용 컴포넌트를 공유함으로써 소프트웨어 재사용을 증가시키고, 개발 및 유지보수 비용을 감소시킬 수 있는 중요한 접근 방법이다. 또한, 프로덕트들 사이의 공통성과 변화성에 초점을 두고 이들에 관한 연구가 진행 중이다. 그 중 아키텍처에서의 변화성은 요구사항 분석·관리 훈련과 아키텍처 개발 사이의 연결을

요구한다. 아키텍처에서의 명백한 설계와, 아키텍처에서 이러한 변화성을 찾는 것을 지원하는 변화점의 정의는 아키텍처 생성자들이 필요로 하지 않는 개발자들이 변화성을 설계하는 것이 가능하다. 그러므로 본 논문에서는 프로덕트들 사이의 공통성과 변화성에 초점을 두고 이들 분류 방법으로 휘처 모델링이라는 개념을 사용하여 분석하였다. 또한 재사용 가능한 아키텍처를 설계하기 위해 변화성의 명확한 표현과 아키텍처에서의 적절한 위치를 식별하기 위해 다양한 변화성 타입을 정의하고 아키텍처의 변화성 표현을 기술하고, 웹기반 교육 시스템 개발에 적용해 보았다. 본 논문에서 제안한 프로덕트 라인을 적용한 웹기반 학습 시스템은 요구사항을 반영하고 기존에 개발되어 사용되고 있는 소프트웨어를 컴포넌트화 하는 작업을 시도하여 이를 재사용하고 통합함으로써 새로운 교육 소프트웨어 개발에 사용하는 일련의 과정에 대하여 연구·기술하였다. 이러한 웹기반 학습 시스템의 가장 큰 장점은 소프트웨어의 재사용성을 확보하는 것이다. 소프트웨어 재사용성 확보는 개발 기간의 단축, 개발 비용의 절약, 생산성 향상, 위험요소의 축소, 향상된 일관성이라는 장점들로 확대된다. 또한 전체 개발단계에서의 복잡도를 감소시키고 대량의 병렬 개발을 지원하며, 시스템의 적응력을 향상과 점진적인 개발을 가능하게 하며 유지보수를 쉽게 한다는 장점을 가진다.

본 논문의 기대효과로는 프로덕트 라인 기반의 웹기반 학습 시스템 응용을 통해서 소프트웨어 품질 및 생산성의 향상이 가능해지고, 핵심자산의 확보를 통하여 다양한 요구사항에 대한 신속한 대응이 가능하다. 그리고 웹기반 학습 시스템 관련 컴포넌트의 확보가 가능해 진다.

향후 연구로는 많은 종류의 컴포넌트들의 개발로 다양한 형태의 웹 학습 시스템의 요구사항에 적절하게 만족시킬 수 있어야 하겠고, 또한 본 논문에서 제안된 프로덕트 라인 아키텍처 및 휘처 모델링을 다양한 응용 영역에 적용시켜 보아야 할 것이다.

참 고 문 헌

[1] J.van Gurp, "On the Notion of variability in Software Product Lines", Proceeding of the Working IEEE/IFIP Conference on software Architecture(WICSA 2001), 2001.

[2] Klaus Schmid, "People Issues in developing Software Product Lines", IESE-Report No. 051.01/E, Version 1.0, 2001.

[3] Northrop, "A Framework for Software Product Line Practice", 2001, http://www.sei.cmu.edu/plp/framework.html.

[4] Paul Clements, Software Product Lines Practices and Patterns, Addison-Wesley, 2002.

[5] John D. Mc Gregor, "Initiating Software Product Lines", IEEE Transactions on Software Engineering, Vol.28, No.7, pp.638-653, July, 2002.

[6] 송재승 외, "Product-Line에서의 Feature-Model의 명세화 방안", 한국정보과학회 춘계학술발표회지, 제29권 제1호., pp. 373-375, 2002.

[7] Kyo C. Kang, "Feature-Oriented Product Line Engineering", IEEE SOFTWARE, Vol.19, No.4, pp.58-65, July/August, 2002.

[8] Daniel Fey, "Feature Modeling : A Meta-Model to Enhance Usability and Usefulness", SPLC 2, San Diego, CA, USA, Vol.2379, pp.198-216, 2002.

[9] Klaus Schmid, "The Economic Impact of Product Line Adoption and Evolution", IEEE SOFTWARE, Vol.19, No.4, pp.50-57, July, 2002.

[10] Charles W. Krueger, "Variation Management for Software Product Lines", SPLC 2, San Diego, CA, USA, Vol.2379, pp. 37-48, 2002.

[11] Van Zyl, J., "Product Line Architecture and the Separation of Concerns", SPLC 2, San Diego, CA, USA, Vol.2379, pp. 90-109, 2002.

[12] Stefan Ferber, "Feature Interaction and Dependencies : Modeling Features for Reengineering a Legacy Product Line", SPLC 2, San Diego, CA, USA, Vol.2379, pp.235-256, August, 2002.

[13] 전주현 외, "컴포넌트를 이용한 웹기반 학습 시스템에 관한 연구", 한국정보교육학회 2001년 하계학술발표 논문집 6권 2호, pp.431-439, 2001.

[14] 전병호, "웹 프로그래밍 학습 시스템 설계 및 구현", 컴퓨터교육학회논문지 5권 3호, pp.69-77, 2002.

[15] Hassan Gomaa, "Modeling software Product Lines with UML", IESE-Report No. 051.01/E, Version 1.0, 2001.

[16] Colin Atkinson, Component-based product line Engineering with UML, Addison-Wesley, 2002.

[17] Lars Geyer and Martin Becker, "On the Influence of Variabilities on the Application-Engineering Process of a Product Family", SPLC 2, San Diego, CA, USA, Vol.2379, pp.1-14, August, 2002.

[18] Michel Jaring, "Representing Variability in Software Product Lines : A Case Study", SPLC 2, San Diego, CA, USA, Vol.2379, pp.15-36, August, 2002.

[19] Steffen Thiel, "Modeling and Using Product Line Variability in Automotive Systems", IEEE SOFTWARE, Vol.19, No.4, pp.66-72, July, 2002.

[20] Andreas Hein, "Systematic Integration of Variability into Product Line Architecture Design", SPLC 2, San Diego, CA, USA, Vol.2379, pp.130-153, August, 2002.

[21] Action Semantics Models, Unified Modeling Language Specification, Version 1.5 OMG document, Formal/03-03-01, 2003.

[22] 김행곤 외, "프로덕트라인 아키텍처상의 컴포넌트 변화성 표현", 한국정보처리학회 소프트웨어공학논문지 제6권 제4호., 2004.

김 행 곤

e-mail : hangkon@cu.ac.kr
1985년 중앙대학교 전자계산학과(공학사)
1987년 중앙대학교 전자계산학과(공학석사)
1991년 중앙대학교 전자계산학과(공학박사)
1987년~1997년 미 항공우주국 객원 연구원
1987년~1989년 한국전기통신공사 전임연구원
1988년~1989년 AT&T 객원 연구원
2001년 Central Michigan University 방문교수
1990년~현재 대구가톨릭대학교 컴퓨터공학과 교수
관심분야 : CBSE, 소프트웨어 재공학, CASE, 유지보수 자동화 툴, 사용자 인터페이스, 요구공학 및 도메인

김 수 연

e-mail : gjtns9384@cu.ac.kr
2002년 경일대학교 컴퓨터공학과 (공학사)
2004년~현재 대구가톨릭대학교 전자계산교육학과 석사과정
관심분야 : CBSE, 소프트웨어 분석·설계, e-learning, 학습컨텐츠 개발