

회차 모델 기반의 웹 학습 컴포넌트 변화성에 관한 연구

A Study On Variability of Web Learning Component Based On Feature Model

민병진(대구가톨릭대학교 컴퓨터정보통신공학부)
김지영(대구가톨릭대학교 컴퓨터정보통신공학부)
김행곤(대구가톨릭대학교 컴퓨터정보통신공학부)

요 약

프로덕트 라인은 다양하게 빠르게 변화하는 시장의 요구사항과 특정 도메인에 속하는 응용들 간의 재사용 가능한 아키텍처 및 컴포넌트의 구성으로부터 연관된 시스템 구축 시 생산성과 품질의 향상을 제공함으로써 현재 많은 관심의 초점이 되고 있다. 이들 프로덕트 라인에서는 프로덕트들 사이의 공통성과 변화성에 초점을 두고 이들 분류 방법으로 회차 모델링이라는 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. 따라서, 명확한 변화성의 표현과 아키텍처에서 변경되는 적절한 위치를 식별하는 것이 중요하다. 그러므로, 본 논문에서는 회차 모델을 기반으로 한 아키텍처 상의 컴포넌트 변화성과 컴포넌트 간의 관련성에서의 변화성 표현 방법을 기술하고, 제시한 이론을 기반으로 웹 학습 시스템을 개발하고자 한다.

I. 서론

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 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].

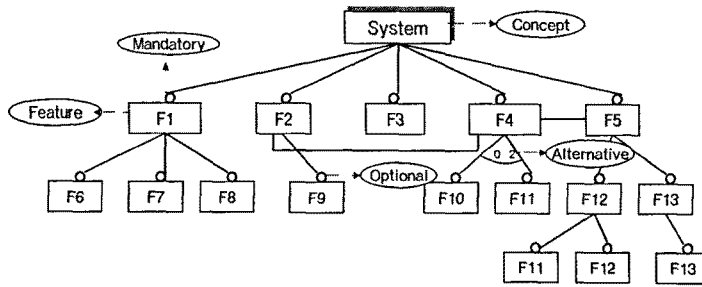
컴포넌트 기반 개발 기술은 많은 연구를 통하여 이미 성숙되어 졌고, 시장 규모면에서 큰 부분을 차지하고 있고, 기술성도 높다고 볼 수 있다. 이에 컴포넌트 기반 개발 기술을 포함하고 있는 프로덕트 라인 기반 개발(Product Line based Development) 기술과 모델 주도형 개발(Model Driven Development) 기술이 새로이 등장하고 있다. 프로덕트 라인 기반 개발 기술의 목표는 일련의 유사한 소프트웨어 시스템의 공통성과 구별되는 특성을 이해하고 제어함으로써 시스템의 체계적인 개발을 지원하는 것이다. 컴포넌트 가변성을 이용하여 사용자가 용도에 맞게 기능을 특화할 수 있고 다양하고 빠르게 변화하는 시장의 요구사항과 특정 도메인에 속하는 응용들 간의 재사용 가능한 아키텍처 및 컴포넌트의 구성으로부터 연관된 시스템 구축 시 생산성과 품질의 향상을 제공함으로써 현재 많은 관심의 초점이 되고 있다[3].

본 논문에서는 프로덕트 라인을 기반으로 회차 모델링을 통한 아키텍처 기술에서의 다양한 변화성 타입을 기술하고, 변화성의 타입을 표기하는 방법과 변경이 지원되는 아키텍처에서 위치를 쉽게 찾기 위해 컴포넌트 변화성에 대해 제시하고, 제시한 이론을 기반으로 웹기반 학습 시스템의 설계 과정을 거쳐 구현하고자 한다.

II. 관련 연구

2.1 회차 모델

회차(Feature) 모델은 프로덕트 라인 요구사항 분석의 핵심 결과로 프로덕트 라인 구성의 기능적, 비기능적인



(그림 1) 다이어그램으로 표현된 휘처 모델

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. (그림 1)은 휘처 모델의 예를 나타낸다[4,5].

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.

휘처의 타입은 아래의 <표 1>과 같이 구분되어진다[6].

<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
대안적 (Alternative)	선택적으로 추가되어질 수 있는 휘처중의 하나가 추가되는 경우의 휘처

2.2 프로덕트 라인의 변화성

동일한 프로덕트 군에 속하는 프로덕트들은 많은 공통성을 가지지만, 또한 프로덕트 사이의 변화성도 있다. 변화성은 다양한 사용자, 다양한 설계와 구현요구사항에 따라 제공되고, 도메인 분석시에 식별된다. 따라서 변화성은 프로덕트 라인 아키텍처 설계 단계에서 고려되어지고, 코드 레벨이 아닌 아키텍처레벨에서 다루어져야 한다. 또한 변화성은 컴포넌트 조립 시에 컴포넌트 행위가 변경될 수 있는 특정 변화점에서 가능하고, 컴포넌트 설계 동안에 결정된다. 변화성은 다음과 같은 범주로 분류된다. 휘처 변화성은 특정 휘처의 정의와 구현에서 변화성, 하드웨어 플랫폼 변화성은 제어기, 메모리, 장비 등의 타입에서 변화성, 성능과 속성 변화성은 요구된 성능과 동시에 지원 및 오류 관리와 같은 속성에서의 변화성을 의미한다[7].

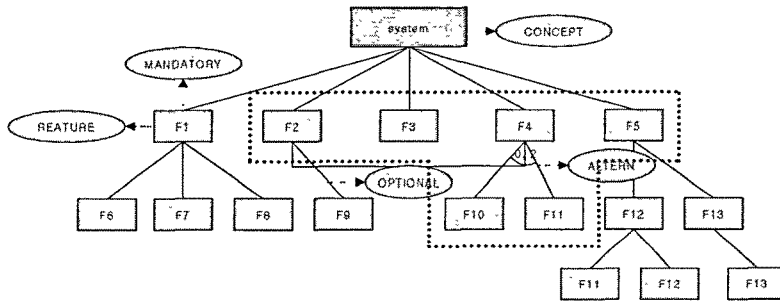
2.3 웹기반 학습 시스템

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 의미있는 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 있다[8].

III. 아키텍처 상의 컴포넌트 변화성

3.1 아키텍처상의 휘처 모델

휘처 모델은 다양한 변화성을 보여주고, 요구사항 사이의 변화성을 만족하기 위해 아키텍처에서의 변화점을 제시하고 어떻게 변화점을 표현할 것인지의 지침을 제공한다. 또한 개발자 측면에서는 다양한 프로덕트 변화의 개발을 유도하고, 사용자 측면에서는 프로덕트 라인의 기능을 쉽게 이해할 수 있도록 도와주는 기능을 제공하기도 한다. 예로 (그림 2)에서 제시된 점선 부분을 통해서 아키텍처 상에서의 변화성을 설계해 보고자 한다.



(그림 2) 아키텍처상의 휘처 모델

3.2 컴포넌트의 변화성 타입

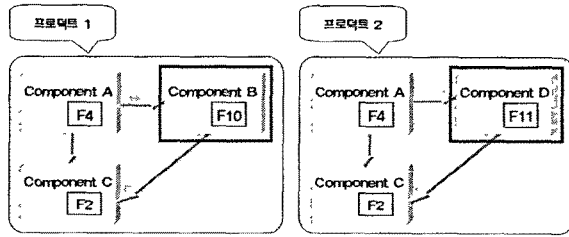
아키텍처에서 다양한 요구사항을 표현하려는 것은 프로덕트 라인 아키텍처에서 프로덕트들 사이의 변화성을 다루기 위해 다양한 선택 사항을 포함하는 경우이다. 이러한 변화성의 타입은 <표 2>와 같이 크게 3 가지 형태로 정의된다.

<표 2> 변화성 타입

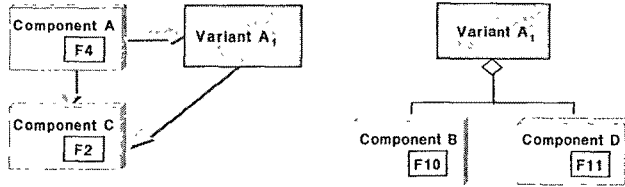
변화성 타입	설 명
Optional	- 어떤 기능이 한 프로덕트에 포함되고 다른 프로덕트에는 불포함 - Optional의 관심은 선택적인 가능성을 갖는 다른 기능들의 관계와 프로덕트에서 포함되지 않은 선택적인 기능인 경우 이러한 관계들이 발생할 수 있음을 나타냄
Alternative	- 추가될 수 있는 여러 선택 중의 하나의 추가
Set of Alternative	- 추가될 수 있는 여러 선택 중의 여러 개가 추가

3.3 컴포넌트의 변화성 표현

하나의 휘처는 하나의 컴포넌트에 대응된다고 가정했을 때 다음과 같이 변화성을 표현해 볼 수 있다. (그림 3)은 두 프로덕트의 컴포넌트 뷰를 보여준다. 이들 컴포넌트 뷰는 (그림 2)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 4)는 하나의 표현으로 두 개의 아키텍처를 표현한다. Variant A₁은 변화점 (VP)을 의미하고, 컴포넌트와 구별하기 위해 1차원의 그레이 박스를 사용한다. Variant A₁의 가능한 구현 기능인 컴포넌트 B와 컴포넌트 D를 추적하기 위해 (그림 5)와 같이 나타낼 수 있다.

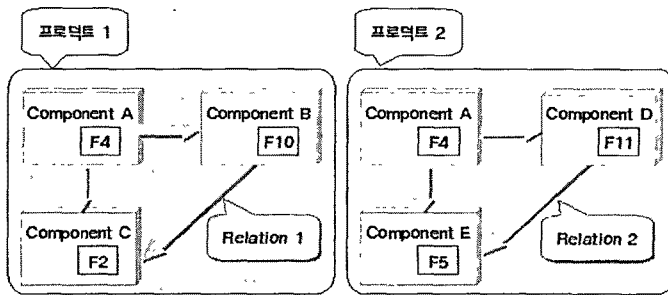


(그림 3) 두개 프로덕트를 위한 선택적인 아키텍처



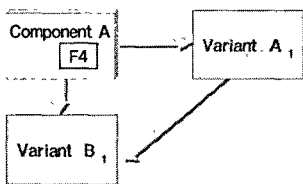
(그림 4) 아키텍처에서의 변화성 (그림 5) 변화성의 선택

변화성은 아키텍처의 컴포넌트 사이에서 관련성을 위하여 발생할 수 있다. (그림 5)은 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 E사이에서 Relation 2를 발생시킨다.

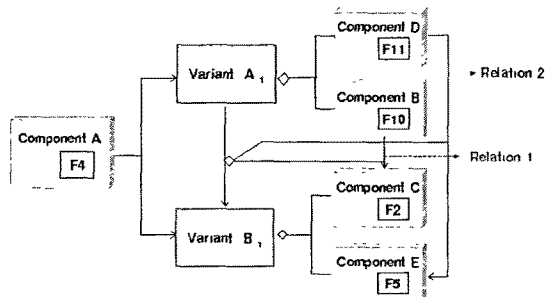


(그림 6) 관련성의 변화성

또한, 이들 관련성에 대한 변화는 연결된 컴포넌트의 변화성을 포함한다. 그러므로 (그림 7)에서와 같이 컴포넌트 C와 컴포넌트 E는 Variant B₁로 표현 가능하다. (그림 8)은 Variant A₁과 Variant B₁ 사이의 의존성을 하나의 큰 그림으로 통합한 것이다.



(그림 7) 아키텍처에서의 변화성

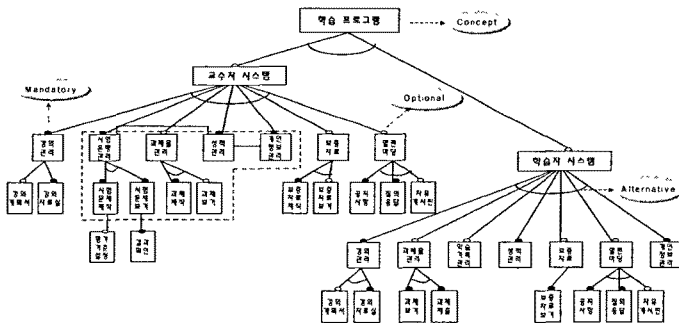


(그림 8) 변화점의 표현

IV. 휘처 모델을 적용한 웹 학습 시스템 개발

4.1 웹 학습 시스템에서의 휘처 모델링

프로덕트 라인의 공통점과 차이점을 분석하기 위한 가장 기본이 되는 휘처 모델은 휘처들 각각의 의미와 성격, 상호연관성을 잘 나타내고 이를 하나의 모델로 집약한 것이다. (그림 9)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. <표 3>은 학습 시스템이 제공하는 교수자 및 학습자 시스템의 상세 기능들을 설명한 것이다.



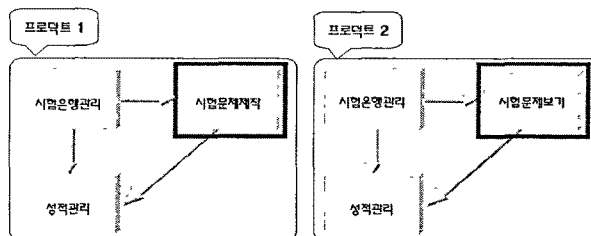
(그림 9) 웹 학습 시스템의 휘처 모델

<표 3> 휘처의 기능 명세

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

4.2 웹 학습 시스템에서의 변화성 표현

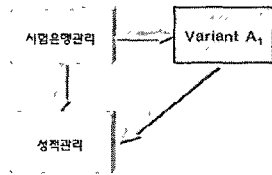
(그림 9)의 휘처 모델링에서 점선 부분의 휘처를 통해 변화성을 제시한다. (그림 10)은 두 프로덕트의 컴포넌트 뷰를 보여준다. 이들 컴포넌트 뷰는 (그림 9)의 점선 부분에서 어떤 휘처를 선택하느냐에 따라 서로 다른 아키텍처의 표현을 나타낸다. 그리고 휘처는 하나의 컴포넌트에 대응된다고 가정한다. (그림 10)에서 두 프로덕트의 차이점은 시험문제제작 컴포넌트와 시험문제보기 컴포넌트의 위치이다.



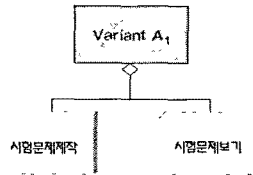
(그림 10) 웹기반 학습 시스템의 두 개 프로덕트를 위한 선택적 아키텍처

이들 변화성을 지원하기 위해 아키텍처는 하나의 다이어그램으로 통합될 수 있다. (그림 11)은 하나의 표현으로 두 개의 아키텍처를 표현한다. Variant A₁은 변화점(VP)을 의미하고, 컴포넌트와 구별하기 위해 1차원의 그레이 박스를 사용한다. 이들 Variant A₁의 가능한 구현 기능을 추적하기 위해 (그림 12)와 같이 나타낼 수 있다. 변화성은 아키텍처의 컴포넌트 사이에서 관련성을 위해 발생할 수 있다.

(그림 13)은 두 프로덕트에 대한 컴포넌트 뷰로 컴포넌트 사이의 관련성에 대한 변화성을 표현한다.

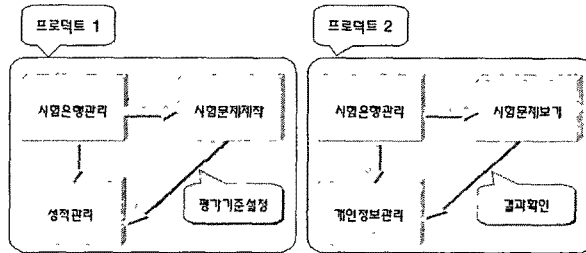


(그림 11) 아키텍처에서의 변화성



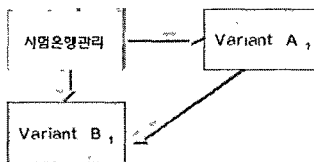
(그림 12) 변화성의 선택

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.



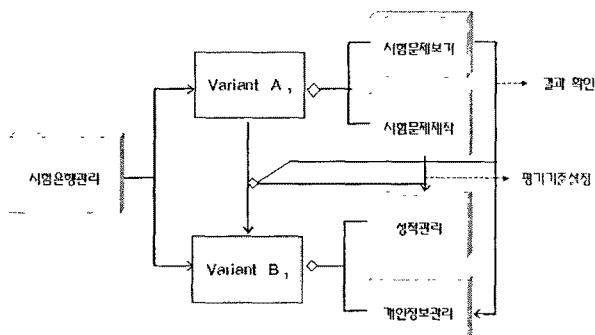
(그림 13) 관련성의 변화성

그러므로 성적관리 컴포넌트와 개인정보관리 컴포넌트는 (그림 14)와 같이 Variant B₁로 표현 가능하다.



(그림 14) 아키텍처에서의 변화성

(그림 15)는 Variant A₁과 Variant B₁ 사이의 의존성을 하나의 큰 그림으로 통합한 것이다.



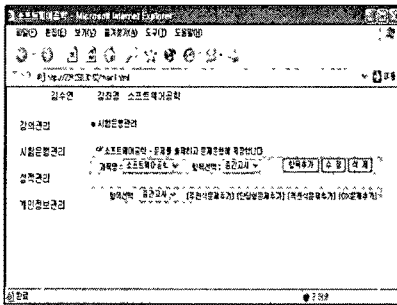
(그림 15) 변화점의 표현

4.3 웹 학습 시스템 구현 예

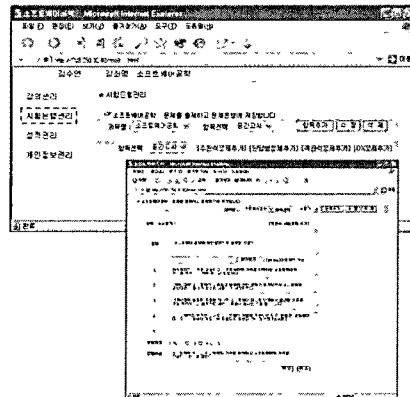
(그림 16)은 교수자 시스템에서 Mandatory 휘처들로 정의된 강의관리, 시험은행관리, 성적관리, 개인정보관리 휘처들로 설계된 웹기반 학습 시스템의 구현 화면을 나타낸 것이다.

(그림 17)은 웹기반 학습 시스템의 시험은행관리 컴포넌트에서의 항목추가 기능을 실행시킨 화면이다.

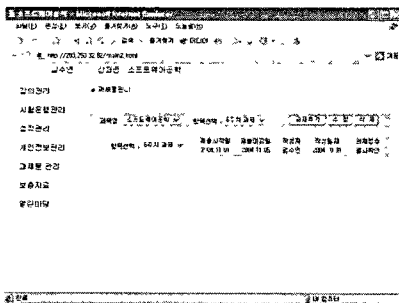
(그림 18)은 교수자 시스템에서 Optional 휘처들로 정의된 과제물 관리, 보충자료, 열린마당 휘처들도 모두 포함시켜 설계된 웹기반 학습 시스템의 구현 화면을 나타낸 것이다. 이렇게 프로덕트 라인의 공통성과 변화성의 특성을 활용하여 공통 도메인 내의 유사한 응용의 개발이 손쉽게 이루어진다.



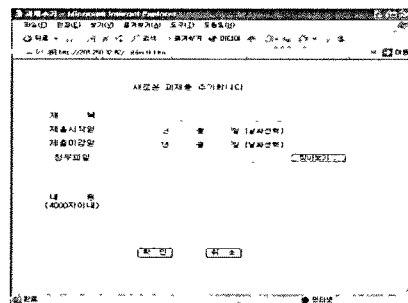
(그림 16) Mandatory 휘처로 설계된 학습 시스템 구현 화면



(그림 17) 시험은행관리 컴포넌트의 항목추가 실행화면



(그림 18) Optional 휘처를 포함시킨 웹기반 학습 시스템 구현 화면



(그림 19) 과제물 관리 컴포넌트의 과제추가 실행화면

V. 결론 및 향후연구

최근 아키텍처와 재사용 컴포넌트들의 공유를 통한 응용 개발이 최상의 패러다임으로 인식되고 있다. 컴포넌트를 기반으로 재사용성과 프로덕트 개발의 생산성, 비용 및 품질 향상을 위해 본 논문에서는 아키텍처에서의 컴포넌트 변화성 표현에 관해 연구했다.

아키텍처에서의 명백한 설계와, 아키텍처에서 이러한 변화성을 찾는 것을 지원하는 변화점의 정의는 아키텍처 설계자들이 필요로 하지 않는 개발자들이 변화성을 설계하는 것이 가능하다. 그러므로 본 논문에서는 프로덕트들 사이의 공통성과 변화성에 초점을 두고 이들 분류 방법으로 휘처 모델이라는 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.

또한, 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.

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 Gorp, "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] 송재승 외, "Product-Line에서의 Feature-Model의 명세화 방안", 한국정보과학회 춘계학술발표회지, 제29권 제1호, pp.373-375, 2002.
- [5] Kyo C. Kang, "Feature-Oriented Product Line Engineering," IEEE SOFTWARE, Vol. 19 No. 4, pp. 58-65, July/August 2002.
- [6] Daniel Fey, "Feature Modeling : A Meta-Model to Enhance Usability and Usefulness," SPLC 2, San Diego, CA, USA, Vol. 2379, pp. 198-216, 2002.
- [7] Charles W. Krueger, "Variation Management for Software Product Lines," SPLC 2, San Diego, CA, USA, Vol. 2379, pp. 37-48, 2002.
- [8] 전병호, "웹 프로그래밍 학습 시스템 설계 및 구현", 컴퓨터교육학회논문지 5권 3호, pp. 69-77, 2002.