

# 확장형UML을 이용한 웹 애플리케이션 모델링 기법1)

박영주<sup>○</sup> 이기열 신우창\* 이병정\*\* 김희천\*\*\* 우치수  
서울대학교 컴퓨터공학부  
서경대학교 인터넷정보학과\*  
서울시립대학교 컴퓨터과학부\*\*\*  
한세대학교 IT학부\*\*  
{ppang<sup>○</sup>, kylee, wuchisu}@selab.snu.ac.kr  
wcshin@imail.skuniv.ac.kr\*  
bjlee@venus.uos.ac.kr\*\*  
khc@hansei.ac.kr\*\*\*

## A Modeling Technique for Web Application with Extended UML

YoungJoo Park<sup>○</sup>, Keeyoull Lee, Woochang Shin, Byungjeong Lee, Heechern Kim, Chisu Wu  
School of Computer Science and Engineering, Seoul National University  
Internet information dept. SeoKyong University\*  
School of Computer Science, University of Seoul\*\*  
Dept. of Computer Engineering, Hansei University\*\*\*

### 요 약

인터넷의 성장과 더불어 웹 환경에서 동작하는 웹 애플리케이션을 체계적으로 개발하고 유지 관리하기 위한 연구가 활발히 진행되고 있다. 이에 웹 애플리케이션 설계 단계에서 웹 애플리케이션을 모델링하기 위한 여러 기법과 프로세스가 제시되었다. 그러나 기존의 연구들은 웹 애플리케이션 모델링의 요구사항에 대한 정확한 정의가 부족하고 웹 애플리케이션의 기술에 의존적이다. 본 연구에서는 웹 애플리케이션 모델링 요구사항에 대한 정의를 명확히 하고 웹 애플리케이션의 기술적인 면에 독립적인 고수준의 추상화를 통해 웹 애플리케이션을 UML로 구조 모델링하는 기법을 보인다. 구조 모델링은 콘텐츠의 구조와 각각의 콘텐츠가 실제 화면에 어떻게 매핑되는지를 표기하여 실제 구현 상태를 모델링에서 나타내는 것이 가능하도록 한다. 마지막으로 제안한 기법을 통해 일반적인 형태의 간단한 웹 애플리케이션을 모델링함으로써 유용성을 검증한다.

### 1. 서 론

인터넷의 성장과 더불어 웹 환경에서 동작하는 웹 애플리케이션을 체계적으로 개발하고 유지 관리하기 위한 연구가 활발히 진행되고 있다. 이에 웹 애플리케이션 설계 단계에서 웹 애플리케이션을 모델링하기 위한 여러 기법과 프로세스가 제시되었다. 그러나 기존의 연구들은 웹 애플리케이션 모델링의 요구사항에 대한 정확한 정의가 부족하고 웹 애플리케이션의 기술에 의존적이다. 본 연구에서는 웹 애플리케이션 모델링 요구사항에 대한 정의를 명확히 하고 웹 애플리케이션의 기술적인 면에 독립적인 고수준의 추상화를 통해 웹 애플리케이션을 UML로 구조 모델링하는 기법을 보인다. 구조 모델링은 콘텐츠의 구조와 각각의 콘텐츠가 실제 화면에 어떻게 매핑되는지를 표기하여 실제 구현 상태를 모델링에서 나타내는 것이 가능하도록 한다. 마지막으로 제안한 기법을 통해 일반적인 형태의 간단한 웹 애플리케이션을 모델링함으로써 유용성을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서 웹 애플리케이션의 특징과 그에 따른 웹 애플리케이션의 요구사항에 대해 알아보고,

3장에서 구조 모델링(structure modeling)의 기술 방법에 대해 설명한다. 4장에서 앞서 설명한 표기법을 이용하여 기본적인 구조를 가진 사이트의 모델링을 실제로 수행한다. 5장에서 웹 애플리케이션을 확장된 UML로 모델링하는 연구 등 기존의 연구들에 대해 알아보고, 본 논문과 장단점을 비교해 보도록 한다. 6장에서 앞으로의 연구방향에 대해 기술한다.

### 2. 웹 애플리케이션의 특성

웹 애플리케이션은 기존의 하이퍼미디어 애플리케이션보다는 더 넓은 개념을 의미한다. Pressman은 전문화된 기능을 가진 웹 사이트와 인터넷 또는 인트라넷, 엑스트라넷에 위치한 정보처리 애플리케이션을 웹 애플리케이션이라 정의하였고, J. Conallen은 사용자의 입력(항해적 입력과 데이터 입력 값)이 상태에 영향을 미치는 웹 시스템으로 정의하고 있다.

웹 애플리케이션의 특징은 다음과 같다.

- 네트워크 상에 위치하므로 누구나 접근이 쉽고, 의견 반영이 신속히 이루어지므로 끊임없이 진화, 갱신된다.
- 외관을 중시하고, 콘텐츠 중심적이므로 구현 작업이 병렬적으로 계획, 진행 될 수 있다.
- 웹은 텍스트, 그래픽, 오디오, 비디오 등의 자료를 사용자에게

1) 본 연구는 한국과학재단 목적기초연구

(R01-2002-000-00135-0)자원으로 수행되었음.

게 제시하기 위해 하이퍼미디어를 사용하는 것이므로 그 자체는 단순한 구조를 가진다. [2]

따라서 웹 애플리케이션의 모델링은 모델 그 자체도 변화에 유연하게 대처할 수 있어야 하고, 병렬적인 계획이 가능해야 할 뿐 아니라, 비기술적인 팀 역시 직관적으로 자신의 업무 흐름을 알 수 있을 만큼 이해하기 쉬워야 한다. 또한 웹 애플리케이션의 분류와 흐름을 가시적으로 표현하여 고객이 스스로의 요구사항을 자연스럽게 표현 할 수 있어야 한다. 세 번째 특징으로 언급한 것과 같이 웹 애플리케이션은 단순한 구조를 가지므로 모델링을 통해 그 구조를 명확히 한다면 구현을 자동화하는 도구의 제작도 가능하다.

웹 애플리케이션 모델은 크게 개념 모델(비즈니스 로직 모델), 항해 모델(네비게이션 모델), 프리젠테이션 모델로 나눌 수 있다. 개념모델은 실제 웹 애플리케이션이 수행해야 할 일들을 순차적으로 표현 해 보는 것으로 일반적으로 유스 케이스 다이어그램(use case diagram)을 이용하여 표현 할 수 있다. 항해 모델은 사용자가 목적하는 페이지로 어떤 경로를 따라 이동할 수 있는지를 노드, 링크, 인덱스, 가이드투어 등으로 나타내는 것을 말한다. 프리젠테이션 모델은 각 엔티티들의 화면배치를 나타낸 것을 말한다. 하지만 개념모델, 항해모델, 프리젠테이션 모델만으로는 위에서 언급한 웹 애플리케이션의 특징들을 반영한 적절한 모델링을 하기에 부족하다고 생각되므로 구조 모델을 제안한다.

3. 구조 모델링(structure modeling) 기법

웹 애플리케이션의 구현 기법이나 기술은 개발자 또는 개발 환경에 따라 달라지므로 어떤 환경에서나 이용할 수 있고, 기술에 독립된 추상 레벨로 웹 애플리케이션의 구조와 기능을 기술할 필요가 있다. 즉 실제로 웹 애플리케이션을 구현하는 자세한 기술을 표기하기 보다는, 목적하는 웹 애플리케이션의 구조와 목적하는 기능에 중점을 맞춘 표기가 필요하다. 따라서 본 논문에서는 화면 구성과 내용을 구조화하여 기술하고 내용과 화면 구성의 매핑 정보를 알려주는 구조 모델을 제안한다. 구조 모델은 각각의 콘텐츠(contents)와 공간을 웹 애플리케이션의 특성에 맞게 구조화 하여 표현 한 것이므로, 개별 콘텐츠를 따로 개발하여 페이지(page)에 삽입하거나 공간에 매핑 할 수 있어서 병렬적인 개발, 컴포넌트(component) 기반 개발의 개념을 충실하게 표현 할 수 있을 뿐만 아니라 빈번한 진화, 갱신을 위한 작업에도 효과적이다. 또한 구조의 표현이므로 기술에 의존적이지 않고, 직관적으로 전체 구조 및 그 표현 형태를 알 수 있으므로 고객이 쉽게 이해할 수 있어서 요구사항 수집 및 피드백(feedback)에 도움이 된다.

웹 애플리케이션은 크게 클라이언트(Client)와 서버(Server)의 두 가지 측면으로 나누어 생각해 볼 수 있다.

클라이언트 쪽은 실제로 구현되었을 때 겉으로 나타나는 것으로, 화면영역(Area), 페이지, 관계 (Relationship)를 표현할 수 있어야 한다. 화면영역은 화면 구성을 뜻하는 것으로 하나의 스크린(Screen)이 보통 두세 개의 프레임(frame)이나 셀(cell)로 나뉘어져 있다. 본 논문에서는 이를 화면영역이라고 통칭한다. 페이지는 하나의 화면영역에 들어가는 실질적인 내용 - 콘텐츠(텍스트, 이미지), 링크(Link), 다이나믹 엔티티(dynamic entity)-를 뜻한다. 페이지는 실제로 존재하는 html 파일을 뜻하는 것이 아니라 하나의 화면영역에 포함되는 변화 가능한 내용이다. 관계에는 화면영역과 화면영역의 관계(include), 페이지와 페이지의 관계(include 또는 link), 화면영역과 페이지의 관계(assign)가 있을 수 있다. 서버 쪽은

사용자의 입력이나 상태의 변화에 따라 애플리케이션을 지원해주는 측면으로 Web server, Html, Java 애플릿, PHP, ASP, CGI, 데이터파일, 데이터베이스 등이 있다.

웹 애플리케이션의 목적은 클라이언트 쪽에서 나타나는 여러 기능들이 의도에 맞게 제대로 동작하는 것이고, 서버 측면의 엔티티는 해당 웹 애플리케이션의 기능을 구현하기 위한 것들로 일반적으로 특정 기술에 의존적인 면이 많다. 따라서 클라이언트 쪽에 표현되는 구조를 주로 표현하고, 필요에 따라 서버에의 요구사항 및 기술들을 첨가하도록 할 수 있겠다.

전체 화면을 어떻게 구성하느냐에 따라 화면은 하나, 혹은 복수개의 화면영역으로 구성된다. 하나의 화면영역은 역시 복수 개의 다른 화면영역들을 포함할 수 있다. 따라서 일반적으로 화면영역들은 포함관계(include)를 가진다.

하나의 화면영역에 들어가는 실질적인 내용을 페이지라고 하는데, 각각의 페이지와 그들 간의 관계를 도식화 할 수 있다. 페이지는 어느 범위까지를 페이지로 보느냐라는 관점에 따라 그 크기가 달라질 수 있는데, 화면영역과 마찬가지로 복수 개의 다른 페이지를 포함(include) 할 수 있고, 그 내용에 따라 다른 페이지를 링크(link)로 불러올 수 있다. 각각의 링크는 실선 위에 스테레오타입을 기술함으로써 그 성격을 명확히 하도록 한다. 다이나믹 콘텐츠 역시 페이지로 표현 할 수 있는데, 해당 다이나믹 콘텐츠가 유효한 기간 동안 같이 유지되는 페이지에 포함관계로 표시한다. 페이지들의 관계(링크 또는 포함)는 기존의 UML을 이용하여 표기하도록 한다.

페이지가 정의되면 각각의 페이지들이 어느 화면영역에 매핑되는지를 점선화살표로 표시해 준다. 이는 전체적으로 웹 애플리케이션이 어떤 식으로 동작하는지 기술할 수 있고, 화면이 어떻게 변화해 가는지를 쉽게 알 수 있는 방법이다.

표 1 표기법

	표기법	의미
공간	화면영역 	변화 가능한 하나의 의미 있는 내용을 담고 있는 공간
	페이지 	변화 가능한 의미 있는 내용 단위
내용	다이나믹 페이지 	각종 다이나믹 엔티티
	포함관계 	화면영역과 화면영역, 페이지와 페이지 간의 포함 관계
관계	링크 	페이지와 페이지 간의 관계
	매핑 	페이지를 화면영역에 매핑

위의 표기법을 이용하여 표기한 구조 모델을 항해 모델, 프리젠테이션 모델과 병행하여 사용할 경우 웹 애플리케이션을 효과적으로 표현 할 수 있다. 구조 모델링을 할 경우 화면의 구성 및 페이지들이 구조화되어 있고, 각각의 화면영역과 페이지 간의 관계도 명확하므로, 개발 시에 이용할 수 있는 자동화 도구를 개발 하는데도 유용하다.

4. 사례 연구

그림1은 쇼핑몰 사이트에서 중요한 기능 중의 하나인 상품구매에 관한 개념모델이다. 본 논문에서는 J. Conallen과 같이 유스 케이스 모델을 단순화 하여 나타내었다. 상품 선택에서부터의 주요한 기능들을 기술 하였다.

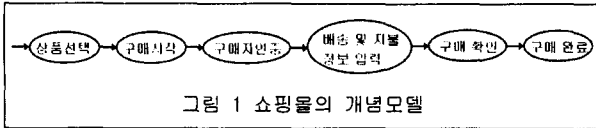


그림 1 쇼핑몰의 개념모델

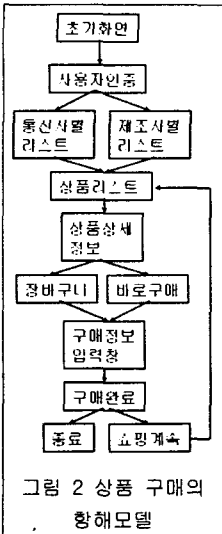


그림 2 상품 구매의 항해모델

그림 2는 사용자가 쇼핑몰을 이용하는 항해경로를 나타낸 것이다. 사용자는 항해모델의 링크를 따라 자신의 목적(원하는 상품의 구매)하는 페이지로 이동 할 수 있다.

그림1과 그림2에서 알 수 있듯이 개념모델과 항해모델만으로 웹 애플리케이션의 설계 및 구현을 표현하기는 힘들다. 그러한 문제를 해결하기 위해 J. Conallen의 연구와 같이 class단위로 표기하게 되면 기술에 의존적일 수밖에 없고 사용자로부터 적절한 요구사항을 알아낼 수 없을 뿐만 아니라 피드백도 힘들다는 문제점을 지니게 된다.

그림3은 일반적으로 많이 이용되는 웹 애플리케이션의 형태를 구조 모델링 한 것이다.

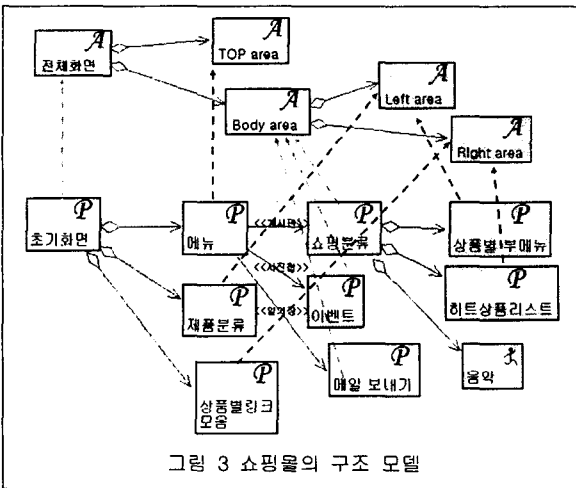


그림 3 쇼핑몰의 구조 모델

우선 전체 화면을 top area와 body area으로 나누고, 다시 body area는 left area와 right area로 나뉜다. 페이지는 메뉴, 제품분류, 상품별 링크는 페이지를 포함하는 초기 화면이 있고, 쇼핑분류를 선택하면 body area에 쇼핑분류 페이지가 뜨게 된다. 쇼핑분류 페이지는 좌측 화면영역의 상품별 부메뉴와 우측 화면영역의 히트상품리스트, 그리고 쇼핑분류 페이지가 실행되는 동안 계속 유지되는 다이나믹 콘텐츠(오디오 프로그램)로 구성된다.

구조 모델에 그림4와 같은 간단한 프리젠테이션 모델을 부가적으로 표현하면 이해도가 훨씬 높아질 것이다.

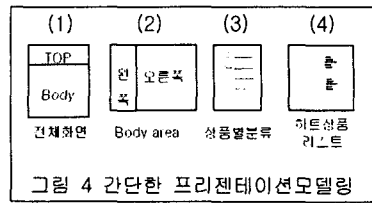


그림 4 간단한 프리젠테이션모델링

더욱 나아가 항해 모델에서 각각의 노드들이 구조 모델의 어떤 페이지들로 구성되는지에 대한 정보가 포함된다면 전체 흐름 및 상황에 따른 화면 구성을 살펴보는 데도 도움이 된다.

5. 관련 연구

웹 애플리케이션 모델링 연구는 모델링의 다양한 측면에서 진행 되고 있다. J. Conallen의 연구는 콘텐츠의 역할보다는 구현 기술(프로그래밍언어)에 따라 다른 기호를 사용하여 기술 방식 자체가 복잡하다. html파일 및 프로그래밍 기술에 의존적이므로 새로운 기술의 경우 적용하기 힘들다는 단점을 가진다. 또한 화면 공간의 사용에 대한 명확한 정의 및 기술 방법에 대한 설명이 부족해서 실제로 프레임간의, 또 타겟 프레임과 내용 사이의 관계를 알기가 힘들고, 하위 수준의 로직한 부분까지 다루게 되므로 직관적으로 그 흐름을 살펴보기 어렵다. Luciano Baresi의 연구는 기존의 UML을 그대로 이용하고, 네비게이션 부분을 확장하여 표현하는 기법을 소개하고 있다. 웹 애플리케이션의 특징인 투어와 인덱스를 적절히 표현할 수 있으나, 저수준의 기능적인 면과 결부되어 실제로 웹의 구동상태를 살펴보는 것은 힘들다. Stefano Ceri의 경우 서버 측면을 강조 하고 있으므로 내부적인 구동 모습을 살펴보기에는 좋으나 그것이 실제로 화면과 어떻게 연결되는지를 알기 힘들기 때문에 초기의 모델링에서 이용하기에는 어려울 것이라 생각된다. Sheldon는 기존의 UML을 단순화 하여 그대로 이용하고 있어서 링크나 assign, 화면 구성과 같은 웹 애플리케이션의 특징을 표현하기 힘들다는 단점을 가진다.

6. 결론

본 논문에서는 웹 애플리케이션을 추상화 하여 그 구조를 모델링하는 기법을 제시한다. 구조 모델링은 기존의 모델링 기법에 비해 좀 더 기술 독립적이며 언어 독립적이다. 또한 가시적 표현을 이용하였으므로 직관적으로 이해 할 수 있어 특별한 지식이 없는 사람도 이용할 수 있으므로 요구사항분석, 병렬적 개발에 용이하다. 또한 이를 항해모델 및 프리젠테이션 모델과 함께 사용함으로써 실제 개발 환경에서 사용될 수 있다. 향후 과제로 지원 도구의 개발과 좀 더 다양한 사례를 통한 방법론의 검증이 필요하다.

7. 참고 문헌

- [1] J. Conallen, *Building Web Applications with UML*, Addison Wesley, 2003.
- [2] R. S. Pressman, *Software Engineering : a Practitioner's Approach* 5th ed., McGraw Hill, 2001.
- [3] Luciano Baresi, Franca Garzotto, and Paolo Paolini, "Extending UML for Modeling Web Applications", *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.
- [4] Stefano Ceri, Piero Fraternali, and Maristella Matera, "Conceptual modeling of Data-Intensive Web Application", *IEEE INTERNET COMPUTING*, 2002.
- [5] Frederick T. Sheldon, Kshamta Jerath, Young-Jik Kwon and Young-Wook Baik, *Case "Study: Implementing a Web Based Auction System using UML and Component-Based Programming"*, *Proceedings of the 26 th Annual International Computer Software and Applications Conference*, 2002.