

# 웹 응용 재공학에 기반한 핵심 애셋의 추출<sup>1</sup>

이광우<sup>o</sup> 이병정 김희천\* 우치수\*\*

서울시립대학교 컴퓨터과학부

한국방송통신대학교 컴퓨터과학과\*

서울대학교 컴퓨터공학부\*\*

freem@intizen.com, bjlee@venus.uos.ac.kr, hckim@knou.ac.kr\*, wuchisu@selab.snu.ac.kr\*\*

## Extracting Core Assets Based on Reengineering Web Applications

Kwangwoo Lee<sup>o</sup>, Byungjeong Lee, Heecheon Kim\*, Chisu Wu\*\*

School of Computer Science, University of Seoul

Dept. of Computer Science, Korea National Open University\*

School of Computer Science, Seoul National University\*\*

### 요 약

오늘날 많은 분야에서 웹 응용이 사용되고 있다. 이러한 웹 응용은 기술의 발전으로 다양한 요구사항을 만족 시키고 있다. 하지만, 중점 요구사항이 기능적인 면과 time-to-market에 한정되고, 이에 웹 응용은 비대해졌으며, 때에 따라 원하는 효과를 얻지 못하기도 한다. 이런 정들로, 웹 응용 재공학의 필요성이 높아졌다. 본 논문에서는 동적으로 생성되는 웹 응용의 재공학에 초점을 두어, 재사용 가능한 핵심 애셋의 효과적인 추출 방법을 제안한다.

### 1. 서 론

오늘날 많은 분야에서 웹 응용(web application)이 사용되고 있고, 이에 웹 응용은 개발에서 유지보수까지 빠르게 연구되고 있다. 이러한 웹 응용의 빠른 발전은 웹 응용의 양적, 질적 향상을 이루었다.

하지만, 웹 응용의 성장은 주로 기능적인 면과 time-to-market이라는 요구사항의 만족에만 초점을 두고 있다. 그에 따라 웹 응용을 개발하기 위해서 많은 비용과 인력 그리고 시간이 투자되고, 그에 반해, 얻는 효과는 크지 못하게 되었다. 이에 점차적으로 웹 응용 재공학의 필요성이 높아져 가고 있다. 이런 웹 응용의 재공학에서 재사용은 매우 유용한 개념중의 하나이다.

본 논문에서는 웹 응용의 재공학에 초점을 두어, 그 중에서도 동적으로 생성되는 웹 응용에서 핵심 애셋(core asset)을 효과적으로 추출하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들에 대해 알아본다. 3장에서는 웹 응용에서 애셋이 어떠한 특성을 지니고 있는지 알아보고, 실제 핵심 애셋의 추출 단계에 대해 논한다. 4장은 실제 적용 사례를 보여주고, 마지막으로 5장 결론에서 요약과 향후 연구 과제를 기술한다.

### 2. 관련 연구

[1]에서는 웹 응용의 Lightweight 프로덕트 라인 재공학 접근

방법을 제시하고 있다. 하지만, 핵심 애셋을 추출하는 과정에서 소스 코드의 일치성(중복)에만 기반하여 핵심 애셋을 추출하고 있다.

[5]에서는 동적으로 생성되는 웹 응용에 초점을 둔다. 이런 형태의 웹 응용은 다수의 중복 요소를 가지며, 구문 분석(parsing)을 통해 중복들을 찾아 특성에 따라 구분 짓고, 각각의 특성에 맞는 해결 방안을 제시하고 있다.

[6]에서는 웹 응용의 재구조화에, 변환 규칙을 적용하고 있다. 실제로 BNF 품을 이용하여 HTML 문서를 재구조화 하는 모습을 보여주고 있으며, 명확한 변환 규칙으로 명확한 결과물을 만들어 내고 있다. 하지만 그 적용 범위가 정적인 페이지로 제한된다.

[7]에서는 XML 형식의 웹 문서를 트리 구조로 표현하고, 트리의 노드가 담고 있는 의미를 기반으로 웹 문서를 재구조화하는 모습을 보여준다. 하지만, 이처럼 웹 응용을 트리 구조로 표현 하였을 때, 모든 노드가 각각 의미를 담고 있는 것은 아니다. 따라서 일반적으로 적용할 수 있는 방법이 필요하다.

### 3. 애 셋

현존하는 많은 웹 응용은 기술적으로 다양한 요구사항을 만족 시키고, 더불어 수요자의 의사를 반영하여 동적인 방식으로 생성되는 형식을 취하고 있다. 이런 방식은 수요자의 요구사항에 일관된 구현을 제공하고 있지만, 제한된 범위의 요구사항 처리 능력과 코드의 중복[5] 등의 문제점을 가지고 있다. 이러한 웹 응용

<sup>1</sup> 본 연구는 한국과학재단 목적기초연구(R01-2002-000-00135-0)지원으로 수행되었음.

을 제공할 한에 있어, 적절한 핵심 애셋의 추출은 폭넓은 요구사항을 수용하고 낭비되는 리소스를 막는데 도움을 준다.

핵심 애셋은 '공통 부분'(common parts)와 '가변 부분'(variable parts)로 나누어 볼 수 있다. 즉, 여러 웹 응용의 공통적인 부분에, 각각의 요구사항을 잘 수용하는 부분을 결합함으로써 하나의 웹 응용이 만들어 지게 된다 [1].

### 3.1 핵심 애셋과 동적 생성 웹 응용

동적으로 생성되는 웹 응용은 일반적으로 '카페'나 '블로그' 같은 커뮤니티 웹 응용에서 쉽게 찾아볼 수 있는 형태이다.

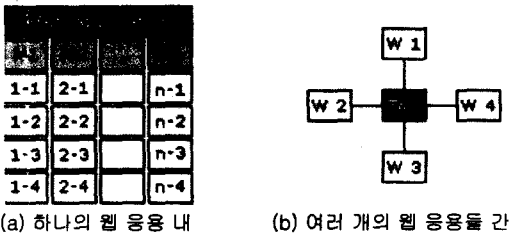


그림 1. 동적 생성 웹 응용의 구성

그림 1은 동적으로 생성되는 웹 응용의 구성을 추상화 한 것이다. 그림 1의 (a)는 한 웹 응용 내에서의 모습이며, M1, M2, ..., Mn은 각각 해당 웹 응용의 메뉴들이다. 각 메뉴에 해당하는 페이지들은 일반적으로 공통의 레이아웃과 기술(technology)을 사용하고 있다. 그림 1의 (b)에서 W1, W2, ... 등은 독립적인 웹 응용이지만, 이 또한 웹 응용 단위로 보았을 때 공통의 기술을 사용하고 있다. 이 경우는 W1, W2 등의 웹 응용을 동적으로 생성, 관리해주는 웹 응용이 따로 있는 형태이다(그림 1의 (b) 중 'Tool').

이런 특성에 의해 동적으로 생성되는 웹 응용은 핵심 애셋의 추출이 용이한 편이다. 그 이유는 이미 하나의 웹 응용이 생성될 때, 요구사항을 수행하는데 기능들을 컴포넌트 단위로 나누어 만들었기 때문이다. 따라서, 컴포넌트를 정제하여 공통 부분과 가변 부분으로 구분하면 그 자체로 훌륭한 핵심 애셋이 될 수 있다.

### 3.2 핵심 애셋의 추출

3.1 절에서 기술한 특성을 지닌 웹 응용에서, 공통 부분과 가변 부분을 적절히 추출해 낼 수 있다면, 핵심 애셋의 재사용 및 관리 측면에서 많은 효과를 얻을 수 있다. 이 과정에서 핵심 애셋이 될 수 있는가의 여부에 의미를 담는 일이 추가되어야 한다. 이렇게 추출된 핵심 애셋은 보다 유용하게 사용될 수 있다.

핵심 애셋의 추출 과정은 다음의 '추출 단계'를 따른다. ① 애셋의 선정을 위한 웹 페이지의 구문 분석, ② 애셋의 의미 분석 및 핵심 애셋의 추출 그리고 ③ 추출된 핵심 애셋의 분류(classification)라는 과정을 거치게 된다.

단계 ①에서 대상이 되는 웹 페이지들을 트리 구조로 구문 분석(parsing)한다. 이렇게 얻어진 트리에서 핵심 애셋의 선정은 다음의 단계 ②에서 적용되는 규칙에 의해 결정된다.

단계 ②에서는 전 단계에서 만들어진 트리 내 요소의 의미를 분석한다. 즉, 중복되는 서브 트리(가공식 트리 전체이거나 노드 일 수 있다, 이하 서브 트리)를 무조건 핵심 애셋으로 추출하는

것이 아니라 그 중 의미 있는 것들만이 핵심 애셋이 되는 것이다.

핵심 애셋으로 추출할 때 일정한 규칙을 따르게 되는데, 그러한 규칙에는 '중복', '기능' 그리고 '표현'이 있다. 첫째로 중복이란, 거의 모든 페이지에 공통적으로 들어가는 정적인 요소이다. 예로, 한 웹 응용내의 모든 페이지에 포함되는 저작권 관련 텍스트를 들 수 있다. 둘째로 기능이란, 기능적인 면을 고려한 것이며, 주로 프로그래밍 언어들이 해당되며, 트리 내에서 중요한 기능을 맡고 있는 서브 트리가 해당된다. 여기서, 여러 애셋간에 다소 코드의 중복이 있다 하더라도 웹 응용 내에서 서로 다른 기능을 가지고 있다면 다른 애셋으로 다루게 된다. 마지막으로 표현이란, 웹 브라우저에서 웹 응용(웹 문서)의 표현에 관련된 서브 트리이며, 예로 테이블 태그 또는 프레임 셋을 이용한 레이아웃등이 이에 해당된다. 이 사항은 중복의 빈도가 낮은 정적인 요소도 핵심 애셋이 될 수 있음을 의미한다.

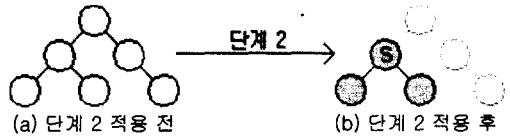


그림 2. 추출 규칙을 적용한 애셋의 선정

그림 2는 이해를 돕기 위한 개념적 그림이며, 구문 분석을 통해 만들어진 트리 가운데 규칙에 해당되는 부분(그림 2의 (b) 중 S 서브 트리)이 하나의 핵심 애셋이 되는 것이다. 핵심 애셋은 그 크기가 되도록 큰 서브 트리가 되도록 한다. 이렇게 추출된 핵심 애셋 중 중복된 것은 하나를 제외하고 삭제한다.

단계 ③에서는 추출된 핵심 애셋들을 비슷한 기능 단위로 묶는다. 그림 3은 핵심 애셋을 추출하는 일련의 과정을 보여주고 있다.

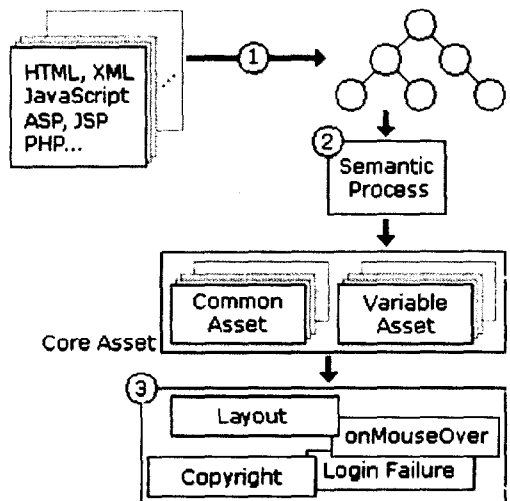


그림 3. 핵심 애셋 추출 과정

### 4. 사례 연구

3.1절에서 기술한 특성을 살펴볼 수 있는 웹 응용 중에서도 본

본문에서는 서울시립대학교(http://www.uos.ac.kr)의 각 학과별 홈페이지에 초점을 맞추어 기술한다. 현재 서울시립대학교의 각 학과별 홈페이지는 자동화 도구에 의해 몇 가지 레이아웃이 제시되고, 필요한 사항을 웹 응용에서 정하고 내용을 입력해주면 자동으로 해당 학과의 홈페이지가 생성된다.

본 연구의 내용을 적용해 보면, 각 학과 홈페이지에 공통적으로 포함되는 컴포넌트인, 메뉴 버튼에 사용되는 자바 스크립트 기능을 하나의 핵심 애셋으로 분류 할 수 있다. 또한 상하의 프레임으로 나누는 기본적인 레이아웃 또한 핵심 애셋으로 분류 할 수 있다. 3.2 절에 언급된 단계에 따라 트리를 그리고 의미를 분석해보면 그림 4와 같다.

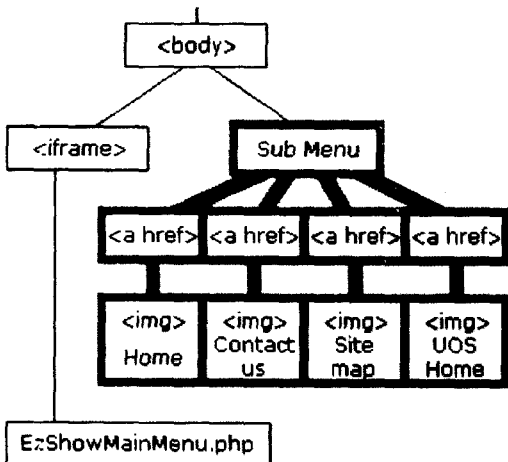


그림 4. Top frame에 들어가는 페이지의 트리 일부

그림 4에서 네 개의 <img> 태그는 실제로 'Home', 'Contact us', 'Site map' 그리고 'UOS Home'으로 링크가 걸려있는 메뉴로, 모든 웹 응용에 일관되게 들어가는 부분이다. 즉, 이것은 추출 단계 ②의 '중복'에 해당하므로, Sub Menu 노드 이하 서브 트리는 하나의 핵심 애셋이 될 수 있다. 이 방법으로 추출된 핵심 애셋을 표로 정리해 보면 표 1과 같다.

표 1. 추출 단계에 의해 추출된 핵심 애셋들

애셋 이름	분류	설명	핵심 애셋
자바스크립트	기능	새 창 열기	Yes / Common
<frame> 태그	레이아웃	전체 레이아웃	Yes / Variable
CSS Style	레이아웃	텍스트 모양 정의	Yes / Variable
학교주소표시	중복	텍스트	Yes / Common
학교 로고	중복	이미지	Yes / Common
...	...	...	...

5. 결 론

현재의 웹 응용들은 코드의 중복으로 자원을 낭비하고 있으며,

이러한 점을 극복하고자, 웹 응용을 재공학하여 핵심 애셋을 추출하는 방법에 대해 살펴보았다. 기존의 연구들은 단순히 코드의 중복을 해결하는 방법만을 제시하고 있으나, 본 논문에서는 트리 구조에 기반하여 의미가 있는 애셋을 추출하는 방법을 제시하였다.

우선, 방법론의 적용 대상은 동적으로 생성되는 웹 응용에 맞추었다. 이런 웹 응용들은 그 성격상 이미 핵심 애셋이 될 수 있는 요소를 다분히 포함하고 있다. 이 특징을 잘 파악한다면 적은 노력으로도 핵심 애셋들을 추출해 낼 수 있다.

핵심 애셋을 추출함에 있어, 본 연구에서는 '추출 규칙'을 적용하였다. 추출 단계는, 구문 분석되어 있는 요소들에 의미적인 부분을 분석하여 핵심 애셋이 될 수 있는지의 여부를 결정하는데 도움을 주는 역할을 한다.

이러한 방식은, 기존의 재사용이라는 측면에도 부합 할 뿐만 아니라 무분별한 핵심 애셋화로 인해, 웹 응용을 유지보수 하면서 일어나는 비용 낭비를 줄일 수 있다.

하지만, 이와 같은 형태는 초점을 맞추었던 웹 응용의 범주를 넘어서면 적절하게 적용되지 않을 수 있다. 또한, 추출 규칙을 적절하게 설정하지 못할 경우 중요한 핵심 애셋을 놓치는 경우가 발생할 수 있다.

따라서, 향후 연구 과제는 보다 폭 넓게 적용될 수 있는 핵심 애셋 추출 아키텍처를 정립하는 것이다. 또한 이러한 추출 방법을 보다 실제적인 웹 응용에 적용해 보아야 한다. 더 나아가서는 이렇게 획득한 핵심 애셋들을 효과적으로 사용할 수 있는 방법을 고려하고, 웹 응용 재공학이라는 측면에서 아키텍처를 정립해야 할 것이다.

6. 참고 문헌

- [1] R. Capilla and J. Duenas, "Light-weight product-lines for evolution and maintenance of Web sites," In *Proc. of the Seventh European Conference On Software Maintenance and Reengineering*, pp. 53 - 62, 2003.
- [2] J. McGregor, L. Northrop, S. Jarrad and K. Pohl "Initiating software product lines," *IEEE Software*, Vol. 19, Issue 4, pp. 24-27, 2002.
- [3] L. Northrop, "SEI's Software Product Line Tenets," *IEEE SOFTWARE*, Vol. 19, Issue 4, pp. 32-40, 2002.
- [4] C. Krueger, "Software product line reuse in practice," In *Proc. of 3<sup>rd</sup> IEEE Symposium Application-Specific Systems and Software Engineering Technology*, pp. 117 - 118, 2000.
- [5] N. Synytskyy, J. Cordy, and T. Dean, "Resolution of Static Clones in Dynamic Web Pages", In *Proc. of 5<sup>th</sup> International Workshop on Web Site Evolution*, 2003.
- [6] F. Ricca, P. Tonella and I. Baxter, "Restructuring Web Application via Transformation Rules," In *Proc. of First IEEE International Workshop Source Code Analysis and Manipulation*, pp. 150 - 160, 2001.
- [7] C. Moh, E. Lim and W. Ng, "Re-engineering structures from Web documents," In *Proc. Of the fifth ACM conference on Digital Libraries*, 2000.