

## 능동적 정보 교환을 위하여 웹 응용을 래퍼 기반 아키텍처로 재공학<sup>1</sup>

이광우<sup>0</sup> 이병정 김희천\* 우치수\*\*

서울시립대학교 컴퓨터과학부

한국방송통신대학교 컴퓨터과학과\*

서울대학교 컴퓨터공학부\*\*

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

### Reengineering Web Applications to Wrapper Based Architecture for Active Information Interchange

Kwangwoo Lee<sup>0</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 and Engineering, Seoul National University\*\*

#### 요 약

오늘날 웹 응용은 다양한 분야에서 사용되고 있다. 하지만, 최근의 웹 응용들은 웹의 가장 기본적인 목적인 정보의 공유를 충분히 만족시켜주지 못하고 있다. 이에 본 논문에서는 웹 응용용한 능동적인 정보의 공유가 이루어 질 수 있도록, 기존의 웹 응용을 래퍼 아키텍처에 기반하여 재공학하는 방법을 제안한다.

#### 1. 서 론

오늘날 웹 응용은 전자상거래, 온라인 커뮤니티, 온라인 교육 등 다양한 분야에서 사용되고 있다. 또한 각 분야의 특성을 구현 할 수 있는 많은 기술들이 연구되고 있다. 이런 웹의 가장 기본적인 공용의 목적은 정보의 공유이다.

하지만, 실제로 웹에서 사용자가 어떠한 정보를 얻기 위해서는 검색엔진들을 통해서 매우 복잡한 경로를 찾아야 한다. 또한 그렇게 찾은 정보라 하더라도, 수시로 이루어지는 업데이트를 사용자 스스로가 확인해야 한다. 이러한 과정은 매우 다양한 주제에 걸쳐 많은 수의 사이트에 해당 되기 때문에, 사용자는 정보의 검색에 대한 부담감을 느끼게 되어, 원활한 정보의 공유가 이루어지지 못하고 있다.

이에, 본 논문에서는 기존의 웹 응용이 정보를 정보 제공자로부터 정보 수집자로 능동적인 흐름을 지닐 수 있도록 래퍼(wrapper-이하 래퍼)를 사용하여 재공학하는 방법을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들에 대해 알아본다. 3장에서는 웹 응용을 재공학하기 위한 웹로그의 특성을 살펴보고, 4장에서는 재공학의 과정을 기술한다. 마지막으로 5장에서는 결론과 향후 연구를 기술한다.

#### 2. 관련 연구

[1]에서는 기존의 웹 사이트의 항해 구조를 컨트롤러 중심의 항해구조로 재구성하는 모습을 보여주고 있다. [2]에서는 웹 공학을 레이아웃과 콘텐츠 그리고 논리(logic)로 분리하여 생각하는 방법은 제안하고 있다. 이는 소프트웨어공학의 기본 개념을 웹 응용의 개별적인 측면에 잘 적용하는 모습을 보여주고 있다. [3]에서는 유사한 구조로 이루어진 웹 문서들의 집합들을 처리하는 프레임워크를 제안하고 있다. 여기에서 컨

텐츠와 함께 구조를 지닌 일련의 웹 문서에 대한 처리를 할 수 있음을 알 수 있다. 하지만 이는 XML 문서에만 적용을 시키고 있다. [4]에서는 웹 응용에서 HTML을 정해진 규칙에 의해서 재작성하는 방법을 제안하고 있다. 하지만 그 적용의 영역이 HTML로 제한되고 있다. [5]에서는 도메인 영역에 맞는 객체 모델을 복원하는 방법을 제시하고 있다. 이는 웹 응용의 역공학시에 외적으로 보이는 모습과 그들간의 관계만이 아닌 데이터의 실제 사용 용도에 따라 웹 응용의 구조를 바라보는 모습을 보여준다 [6]에서는 기존 웹 문서에서 문서의 구조를 HTML에서 유효한 XML문서로 변환하는 방법에 대해 논하고 있다. [7]에서는 웹 응용의 순공학시 효과적인 모델링을 위한 UML의 확장을 보여주고 있다.

#### 3. 웹 응용의 능동적 정보 공유를 위한 웹로그 응용

웹로그 응용(weblog-이하 웹로그)은 수동적으로 정보를 제공하는 웹 응용에 대한 대안으로 제안되고 빠르게 확산되고 있다. 웹로그는 기본적으로 로그(log)를 통한 정보의 부각, XML 기반의 RSS(RDF Site Summary-이하 RSS)명세 [9, 10]를 통한 일관된 형식의 정보의 제공 그리고 트랙백(Trackback-이하 트랙백)으로 보다 양질의 정보 공유를 추구하고 있다.

기존의 웹 응용과 웹로그가 사용자에게 정보를 보여주는 일련의 구조는 그림 1과 같다. 웹로그와 기존의 웹 응용과의 가장 큰 차이점은, 사용자가 웹 응용에게 페이지를 요청할 때, 최근 작성된 글들의 로그를 기반으로 가장 최근의 정보들을 부각시킨 페이지를 보여주는 것이다. 이를 통해 최신 업데이트 정보에 대한 검색의 노력을 줄일 수 있고, 응용간의 로그의 공유를 통해 능동적인 정보 공유가 가능해진다. 여기서 로그 파일은 콘텐츠에 대한 내용을 시간 순서대로 기록해 놓게 되며, W3C에서 제안한 RSS 명세를 따라 만들어진다. 그 후 사용자가 다른 형태의 요청을 할 경우 그에 맞게 질의기를 통해 추출해온 데이터를 보여준다.

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

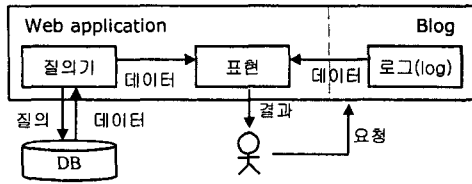


그림 1. 기존 웹 응용과 웹로그 시스템 구조의 컨셉 다이어그램

#### 4. 웹 응용의 재공학 과정

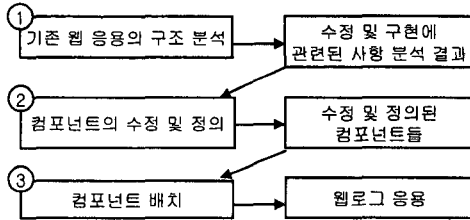


그림 2. 재공학 과정

앞에서 살펴본 특성에 따라 우리는 기존의 웹 응용의 구조를 크게 변경하지 않으면서, 웹로그로 전환 할 수 있음을 알 수 있다.

재공학 과정은 그림 2와 같으며, 크게 3 단계로 이루어진다. 첫 번째는 기존 웹 응용의 구조를 분석하는 것이다. 이 단계에선 기존 웹 응용의 분석을 통해 콘텐츠의 추가/삭제/수정에 관련된 부분과 외부와의 통신이 이루어지는 부분을 찾아낸다. 다음으로는 웹로그의 특성에 맞게 컴포넌트들을 수정 또는 새로 정의하는 것이다. 마지막으로 수정되고 새로 정의된 컴포넌트들을 재배치한다.

본 논문에서는 재공학의 예로 많은 웹 응용의 일반적인 형태인, DB를 사용하는 게시판 형식의 웹 응용을 웹로그로 전환하는 방법에 대해 기술하며, 지면관계상 화면 표현(presentation)에 관련된 부분은 언급하지 않겠다.

#### 4.1 기존 웹 응용의 구조 분석

이 단계에서는 기존 웹 응용의 구조 분석을 통해 웹로그의 특성을 적용시켜야 될 부분을 찾아낸다. 이 과정은 정적 분석과 동적 분석으로 구분된다 [8]. 정적 분석에서는 기본적으로 웹 응용 내부에 있는 컴포넌트간의 정적인 관계를 분석한다. 예를 들면, 웹 페이지 간의 submit 관계 같은 것이 이에 해당한다. 이러한 정적 분석의 결과를 기반으로 동적 분석을 하게 된다. 이는 컴포넌트에 포함된 코드들이 실행될 때에만 그 결과가 나오는 부분에 대한 분석이다. 즉, 컴포넌트간에 일어나는 실제적인 상호작용을 살펴봄을 통해서 어떤 역할을 하는 컴포넌트인지를 알아내는 것이다. 이 과정은 개발자가 개입하여 검토해야 하는 사항이다.

앞의 과정을 통해, 콘텐츠의 추가/삭제/수정에 따라 로그의 내용이 변경되어야 하므로, 기존 웹 응용에서 이와 같은 역할을 담당하는 컴포넌트를 찾아낸다. 또한 트랙백의 지원을 위해 외부와의 통신이 일어나는 컴포넌트들을 찾아낸다.

#### 4.2 컴포넌트의 수정 및 필요 컴포넌트 정의

##### 4.2.1 로그 생성 컴포넌트의 정의

웹로그의 핵심 개념인 로그를 생성시켜주는 컴포넌트는 다음의 두 가지의 경우를 고려하여야 한다. 웹 응용에 담겨져 있는 콘텐츠가 DB의 테이블에 저장되는 경우와, 일반적인 flat file로 저장되는 경우이다.

보통의 게시판 형식은 전자의 경우이며, 웹 응용의 해당 데이터를 담는

DB의 구조를 토대로 로그를 생성시켜주는 컴포넌트를 정의한다. 기존의 웹 응용은 HTML의 입력 form들을 통해 사용자가 입력한 콘텐츠를 해당 DB의 table에 저장한다. 이는, 반대로 생각하면 해당 DB의 각 table의 한 행은 로그내의 RSS 명세에 따라 각 <item> 요소들이 될 수 있다. 표 1은 DB의 각 table과 <item> 요소간의 관계를 일부 보여주고 있다. 즉, 기존에 만들어져 있는 해당 질의기를 사용하여 필요한 내용들을 DB로부터 가져와 RSS의 명세에 맞게 로그 파일을 생성시켜주는 컴포넌트를 정의하면 된다.

표 1. DB 와 RSS 간 전이의 예

DB table	RSS <item> 요소
글제목	<title></title>
해당 콘텐츠의 DB 주소	<link></link>
해당 콘텐츠의 내용	<description></description>
해당 콘텐츠의 글쓴이	<author></author>

두 번째로, 웹 응용 내에 콘텐츠들이 flat file의 형태를 띄고 있는 경우이다. 하지만, 대부분의 게시판 형식의 웹 응용이 DB를 사용하고 있고, flat file로 저장되는 경우는 DB에 저장되는 콘텐츠의 일부(이미지, 동영상, 플래시등)이므로 <item>요소의 <description>에 그 정보가 포함된다. 하지만, 부분적으로 로그 구조에 필요한 사항들을 추출해 내야 할 필요가 있는데, 예를 들면, 로그 파일 내에는 해당 웹 응용을 대표하는 이름(예를 들어 'SE 연구실 홈페이지' 등)을 나타내는 <channel> 요소 내의 <title>이 있다. 이 경우는, 일반적으로 웹 응용으로 접근할 때 처음 접하게 되는 페이지(보통, index페이지 또는 home페이지)의 HTML 태그 중 <head> 태그 내의 <title> 태그에 기록이 되어 있다. 하지만 실제 웹 응용의 파일 내에서 명백히 나타나 있지 않은 경우도 있고, 있다 하더라도 그 위치가 정확하지 않을 수 있으므로 개발자가 검토해야 한다.

그림 3은 이에 따라 생성된 로그 생성 컴포넌트의 다이어그램의 일부이다. write\_ok는 기존의 웹 응용에서 새로운 콘텐츠의 추가를 담당하는 컴포넌트이며, 필요한 메소드와 속성을 추가하였다. 또한, 로그를 작성하는 새로운 컴포넌트인 Generate\_Log를 정의하여 새로운 콘텐츠의 추가 시 로그를 생성하도록 설계하였다. 이와 같은 모습은 콘텐츠의 삭제/수정에도 적용된다.

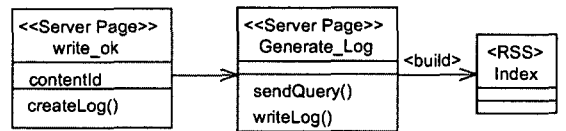


그림 3. 로그 생성 컴포넌트 클래스 다이어그램의 일부

##### 4.2.2 트랙백 처리 컴포넌트의 정의

웹로그의 가장 큰 특징 중의 하나는 트랙백이다. 이는 그림 4와 같이 이미 게시판 콘텐츠에 관련된 정보를 자신의 웹로그에 올리고, 관련된 글이 있다는 정보(URL)만을 원래의 콘텐츠에 알려주는 형식을 취하고 있다. 따라서, 기존의 웹 응용으로부터 외부로의 연결이 트랙백과 관계되었을 때 이를 처리해주는 컴포넌트를 정의해야 한다.

이 경우도 두 가지로 생각해 볼 수 있는데, 외부로 트랙백 주소를 보내는 경우, 외부에서 트랙백 주소가 들어오는 경우이다. 이런 트랙백 처리는 기존의 웹 응용이 외부와 통신을 하는 경우, 그 통신의 형태가 논리에 관련된 부분인지, 아니면 트랙백에 관련된 부분인지를 파악하여 이루어 지도록 하면 된다.

마찬가지로 그림 5는 트랙백 처리에 대한 클래스 다이어그램의 일부를 보여준다. comment\_ok는 기존 웹 응용에서 원본글에 주석글을 기록하는 컴포넌트이며, 트랙백 처리를 위해 check\_trackback과

send\_trackback이 새로 정의 되었다. 우선, 외부와의 통신은 모두 check\_trackback을 거치게 된다. 이 때, 외부에서 트랙백 주소가 들어오는 경우는 웹 응용이 가지고 있는 컨텐츠의 유일한 URL과 구현 당시의 트랙백 처리에 관련된 속성을 가지고 들어오므로, 이를 통해 트랙백 주소임을 감지하고 해당 요청을 트랙백 추가에 알맞게 처리하면 된다. 반대로 외부로 트랙백 주소가 나가는 경우는, 마찬가지로 check\_trackback을 통해 트랙백 전달을 담당하는 send\_trackback을 통해 외부 웹로그에 전달하게 된다.

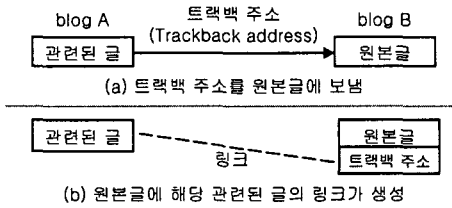


그림 4. 트랙백의 구조

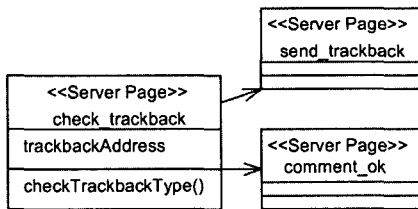


그림 5. 트랙백 처리 컴포넌트 클래스 다이어그램의 일부

4.3 컴포넌트 배치

수정/생성된 컴포넌트들은 기존의 웹 응용과 함께 래퍼 아키텍처에 알맞게 배치되며, 배치를 마친 후 최종적인 웹로그의 시스템 아키텍처는 그림 6과 같다.

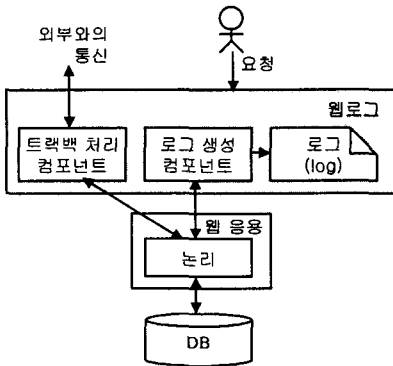


그림 6. 재공학 후 웹로그 시스템 아키텍처

새로 정의된 컴포넌트들은 기존 웹 응용 계층 위에 배치되며, 기존의 외부와의 통신과 사용자의 요청은 이 계층을 통해서 기존의 웹 응용에 전달되게 된다. 기존의 웹 응용은 해당 요청에 따라 컨텐츠에 대한 정보를 DB로부터 얻어온다.

5. 결론 및 향후 연구

오늘날의 웹 응용은 그 역할이 매우 다양하며, 그 중요성도 높아지고 있다. 그런 가운데에서도 변하지 않는 기본 개념은 정보의 공유이다. 하지만, 현재의 방식은, 사용자가 정보를 얻기 위해서 매우 많은 시간과 노력을 들여야 했다. 이런 것에 대한 대안으로 제안된 방식인 웹로그는 로그를 이용하여 사용자에게 가장 최신의 정보를 제공하며, 로그를 RSS 명세에 따라 만들어 사용자에게 보다 능동적인 정보의 제공을 추구할 수 있도록 하였다. 이에 본 논문에서는 DB기반의 웹 응용이 웹로그의 형태로 전이할 수 있는 방법을 논하였다.

웹로그는 로그, RSS 그리고 트랙백을 핵심 기술로 사용하고 있다. 즉, 로그를 처리할 수 있는 방안과 트랙백을 처리할 수 있는 방안을 기존의 웹 응용에 추가해야 한다. 이는 기존 웹 응용에서 이 기능들을 추가할 곳을 분석하고, 두 기능에 관련된 컴포넌트들을 수정/생성한 후, 기존의 웹 응용과 함께 재배포함으로써 해결된다.

이와 같이 웹 응용의 웹로그로의 전이를 통해 정보 공급자는 저비용으로 효율적인 정보 제공을 할 수 있고, 정보 수집자는 적은 노력으로 최신의 정보들을 접할 수 있다. 또한, 래퍼 아키텍처를 기반으로 한 재공학은 비용 측면에서 효과적이다.

앞으로 진행되어야 할 부분은 주어진 컨텐츠를 사용자에게 보여주는 표현(presentation)과 관련된 부분에 대한 재공학 방법을 정립하는 것이다. 또한, 웹 응용의 게시판을 통해 DB 내에 저장된 컨텐츠에 대한 부분만이 아닌, 보다 다양한 논리에 관련되어 생성되는 컨텐츠들에 대해 웹로그로 전환하는 방법에 대한 연구가 필요하다.

6. 참고 문헌

- [1] Y. Ping, and K. Kontogiannis, " Refactoring web sites to the controller-centric architecture," *In Proc. of the 8th European Conference on Software Maintenance and Reengineering*, pp. 204 - 213, 2004.
- [2] E. Kirda, " Engineering Device-Independent Web Service" , PhD thesis, Technical University of Vienna, 2002.
- [3] C. Moh, E. Lim and W. Ng, " Re-engineering structures from Web documents," *In Proc. of the 5th ACM conference on Digital Libraries*, 2000.
- [4] F. Ricca, P. Tonella and I. Baxter, " Restructuring Web Application via Transformation Rules," *In Proc. of the 1st IEEE International Workshop on Source Code Analysis and Manipulation*, pp. 150 -160, 2001.
- [5] G. A. D. Lucca, A. R. Fasolino, P. Tramontana, and U. De Carlini, " Recovering Business Object Model from Web Applications," *In. Proc. of the 27th Computer Software and Applications Conference*, pp.348 - 353, 2003.
- [6] C. Y. Chung, M. Gertz, N. Sundaresan, " Reverse Engineering for Web Data: From Visual to Semantic Structure," *In Proc. of the 18th International Conference on Data Engineering*, 2002.
- [7] J. Conallen, *Building Web Applications with UML 2nd Edition*, Addison Wesley Publishing Company, Reading, MA., 2003.
- [8] A. D. Lucia, R. Francese, G. Scanniello, G. Tortora, " Reengineering Web Applications Based on Cloned Pattern Analysis" , *In Proc. of the 12th IEEE International Workshop on Program Comprehension*, pp. 132-141, 2004.
- [9] World Wide Web Consortium(W3C), " RSS 1.0 specification" , <http://web.resource.org/rss/1.0/spec>, 2000.
- [10] Technology at Harvard Law, " RSS 2.0 specification" , <http://blogs.law.harvard.edu/tech/rss>, 2003.