

# 웹 응용 설계를 위한 패턴 추출 방법+

한지윤<sup>o</sup> 이기열 이병정\* 김희천\*\* 우치수  
서울대학교 컴퓨터공학부  
서울시립대학교 컴퓨터과학부\*  
한세대학교 IT학부\*\*  
{june<sup>o</sup>, kylee, wuchisu}@selab.snu.ac.kr  
bjlee@venus.uos.ac.kr\*  
khc@hansei.ac.kr\*\*

## A Pattern Extraction Approach for Web Application Design

Jiyoon Han<sup>o</sup> Keeyoull Lee ByungJeong Lee\* Heecheon Kim\*\* Chisu Wu  
Dept. of Computer Science and Engineering, Seoul National University  
School of Computer Science, University of Seoul\*  
Dept. of Computer Engineering, Hansei University\*\*

### 요 약

인터넷의 비약적인 발전은 웹 환경으로의 빠른 전환을 요구하고 있으며 웹 애플리케이션의 설계, 구현을 위한 체계적인 개발 방법론이 필요하게 되었다. 기존의 웹 관련 기술들은 웹 애플리케이션을 구현하기 위한 도구와 기반 실행 환경을 중심으로 개발되어 왔다. 하지만 설계 수준과 요구사항 수준에서 웹 개발을 지원하는 체계적인 방법론에 대한 연구는 아직 활발하지 못하다. 재사용성과 견고성을 높이기 위한 방법론으로서 웹 애플리케이션 패턴이 제안되고 있으나, 그러한 패턴을 추출하는 데 있어 정형적인 방법론은 제안되지 못한 상태이다. 본 논문에서는 웹 애플리케이션을 위한 설계 프레임워크에 관한 연구의 세부 과제로서 웹 애플리케이션을 설계하는데 사용되는 웹 패턴을 추출하는 체계적인 접근법을 기술하도록 한다.

### 1. 서 론

인터넷의 비약적인 발전은 웹 환경으로의 빠른 전환을 요구하고 있다. 이에 따라 웹 애플리케이션의 설계, 구현을 위한 체계적인 개발 방법론이 필요하게 되었다. 최근 설계 수준과 요구사항 수준에서 웹 개발을 지원하는 체계적인 방법론에 대하여 연구가 진행되었으며, 그 일환으로 웹 애플리케이션을 모델링하는 데 있어 비즈니스 로직 계층, 항해 계층, 프리젠테이션 계층 등에 적용되는 패턴에 관한 연구가 진행되고 있다[1, 2, 3, 4]. 사용자 업무의 신속하고 효율적인 달성을 위하여 효과적인 항해 경로를 찾는 것은 특히 중요하다. 효과적인 항해 경로를 찾는 작업은 웹 애플리케이션 안의 정보들이 만들어내는 초월 공간(hyperspace)의 크기를 줄여주는 역할을 한다[5]. 따라서 항해 경로 단계에서의 패턴이 활발히 제안되고 있다[5, 6]. 그러나 웹 애플리케이션에 적용되는 항해 패턴을 추출하는 정형화된 프로세스에 대한 연구는 아직 미흡한 실정이다. 본 연구에서는 웹 애플리케이션을 위한 설계 프레임워크에 관한 연구의 세부 과제로서 웹 애플리케이션의 집합을 그 입력으로 하여 그들이 공통적으로 가지고 있는 패턴을 찾아내는 프로세스를 제안하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 애플리케이션의 모델링과 관련된 연구를 살펴보고, 3장에서는 패턴 추출 대

상인 웹 애플리케이션을 정의하며, 4장에서는 웹 패턴을 추출하는 프로세스와 알고리즘을 기술하고 사례를 들어보도록 하겠다. 마지막으로 5장에서는 앞으로 연구가 진행될 방향에 대하여 제시하도록 하겠다.

### 2. 관련 연구

웹 애플리케이션을 모델링하기 위한 연구는 크게 웹 애플리케이션을 객체 지향 개념으로 표현하기 위한 방법론[7]과 웹 애플리케이션의 모델링 기법 자체에 대한 방법론[2, 3, 4]으로 나눌 수 있다. [7]의 연구에서는 UML을 확장하여 웹 애플리케이션을 모델링하는 방법에 대해서 기술하고 있다. 스테레오 타입과 태그드 밸류(tagged value), OCL 형식으로 기술된 제약 조건을 사용하여, 기존의 객체 지향 모델링 개념에 포함되지 않았던 웹 애플리케이션이 가진 하이퍼링크와 페이지, 동적인 내용 구성의 특성을 반영하였다. [2]의 연구에서는 웹 애플리케이션의 한 종류인 하이퍼미디어 애플리케이션을 모델링 하기 위하여 개념적 디자인, 항해 디자인, 추상적 인터페이스 디자인과 그 구현의 네 가지 계층을 소개하고 있다. [3, 4]의 연구에서는 OOHDM의 각각의 개념적 계층에서 재사용 가능한 패턴을 제안하고 있다. 첫째, 객체 지향 개념을 적용하여 하이퍼미디어 시스템 즉 하이퍼미디어 애플리케이션을 만들 수 있는 환경을 설계하는 데 필요한 디자인 패턴을 제안하고, 둘째, 노드와 링크를 만드는 등의 웹 애플리케이션을 모델링하는 데 있어 필요한 항해 계층에서 패턴을 기술한다. 마지막으로 웹 애플리케이션에서 사용자가 가능한 정보를 충분히 받을 수 있도록 하는 GUI 패턴을 제안한다. [7]의 연구에서

+ 본 연구는 한국과학재단 목적기초연구(R01-2002-000-00135-0)지원으로 수행되었음.

는 웹 애플리케이션을 전위, 비즈니스 로직, 후위의 세 계층으로 나누었을 때 UML과 OCL 형태의 제약 사항을 이용하여 웹 비즈니스 로직 계층에 적용할 수 있는 패턴을 제안하고 있다. 전위 계층은 비즈니스 로직으로부터 넘겨진 정보를 사용자의 요구에 맞는 형태로 가공하여 사용자에게 보여주는 역할을 한다. 비즈니스 로직 계층은 후위계층으로부터 데이터와 정보를 모아서 전위로 넘겨주는 역할을 한다. 후위 계층은 데이터 소스를 관리하며 비즈니스 로직 계층으로 요청을 받아 필요한 정보를 비즈니스 로직 계층에 넘겨주는 역할을 한다.

3. 웹 애플리케이션 모델

웹 애플리케이션은 기존의 하이퍼미디어, 클라이언트/서버 스크립트 언어와 미들웨어 기반 컴포넌트까지를 모두 포함하는 복합적인 개념이다. [1]의 연구에서 웹 애플리케이션을 사용자의 입력(항해 개념과 데이터 입력값)이 현 상태에 영향을 미치는 웹 시스템으로 정의하였다. 이러한 정의로서 웹 애플리케이션을 웹 시스템에 의하여 전달되는 비즈니스 로직 상태를 가진 소프트웨어로 생각할 수 있으며, 항해 계층과 프레젠테이션 계층을 분리하여 생각할 수 있게 하였다.

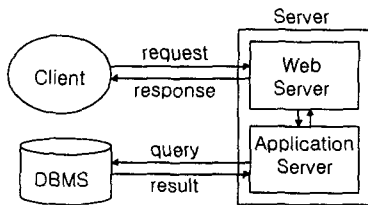


그림 1 웹 애플리케이션 동작 모델

전형적인 웹 애플리케이션의 동작은 그림 1과 같은 일련의 과정으로 이루어져 있다. 사용자가 웹 서버에 URL과 사용자 입력사항을 덧붙여 요청(request)을 하면 웹 서버는 단순히 해당 하이퍼미디어파일이나 멀티미디어 파일을 클라이언트에 전달해 주거나 추가적인 비즈니스 로직을 처리할 필요가 있을 때 해당 애플리케이션 서버에 다시 요청을 의뢰한다. 이 때 그 요청의 성격에 따라 데이터베이스 관리 시스템과 관련 없이 요청이 수행되기도 하고, 데이터베이스 관리 시스템에 질의(query)를 보내고 그 결과(result)를 애플리케이션 서버에서 적절한 처리를 한 후에 클라이언트에게 보내주게(response)된다. 이러한 한 주기 단위의 과정을 액션이라고 정의하도록 하겠다. 액션은 현재의 위치를 파악하기 위한 현재의 위치(Current\_page), 요청(Request)과 응답(Response), 질의(Query)와 결과(Result), 사용자의 계정과 같은 상태 정보를 저장하는 세션(Session), 응답 페이지(Response\_Page)로 이루어진다.

정의 1.

Action = <Current\_Page, Request, Response, Query, Result, Session, Response\_Page>

항해 경로는 일련의 연속된 액션들로 이루어진다. 사용자가 위치할 수 있는 페이지를 각각 하나의 노드로 보고 하나의 액션에 의해서 현재 페이지에서 다음 페이지로 이동하는 것을 간선(edge)으로 표현할 때 항해 경로는 다음과 같이 방향성을 갖는 그래프(directed graph)로 정의된다.

정의 2.

애플리케이션 A = (N, E),

(N = 노드들의 집합, E = 간선들의 집합)

항해 경로 P = (a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., a<sub>n</sub>)

= (n<sub>1</sub>, e<sub>1</sub>, n<sub>2</sub>, e<sub>2</sub>, ..., e<sub>n-1</sub>, n<sub>n</sub>)

(단, n<sub>1</sub>, n<sub>2</sub>, ..., n<sub>n</sub> ∈ N, e<sub>1</sub>, e<sub>2</sub>, e<sub>3</sub>, ..., e<sub>n-1</sub> ∈ E, a<sub>n</sub>은 액션,

(a<sub>n</sub>, a<sub>n+1</sub>) = (n<sub>n</sub>, e<sub>n</sub>, n<sub>n+1</sub>, e<sub>n+1</sub>, n<sub>n+2</sub>) 일 때,

a<sub>n</sub> = <p<sub>n</sub>, ..., p<sub>n+1</sub>>, a<sub>n+1</sub> = <p<sub>n+1</sub>, ..., p<sub>n+2</sub>>.

p<sub>1</sub> = n<sub>1</sub>, p<sub>n</sub> = n<sub>n</sub>)

그림 2에서 P<sub>1</sub>은 n<sub>1</sub>, n<sub>2</sub>, n<sub>4</sub>, n<sub>7</sub>을 거쳐 n<sub>8</sub>을 최종 목적으로 하는 항해 경로를 보여주고 있다. P<sub>2</sub>는 n<sub>1</sub>, n<sub>3</sub>을 거쳐 n<sub>6</sub>를 최종 목적으로 하는 항해 경로를 보여주고 있다.

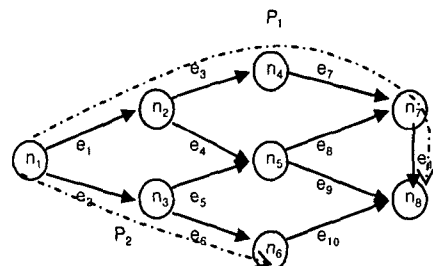


그림 2 웹 애플리케이션의 항해 경로

웹 애플리케이션의 항해 경로 중에서 여러 사용자에게 의해 계속 반복되는 경우 그것을 하나의 후보 항해 패턴으로 제안할 수 있을 것이다. 특히 그러한 후보 항해 패턴이 하나의 애플리케이션에서 뿐 아니라 다른 웹 애플리케이션에서도 반복되는 현상이 나타날 때, 그것을 항해 패턴으로 제안할 수 있다.

4. 웹 애플리케이션 패턴 추출

4.1. 프로세스

사용자가 웹 서버에 접속하여 여러 요청들을 수행하고 나면 웹 서버에는 로그가 남게 된다. 이 로그를 토대로 하여 액션들을 찾아내도록 하겠다. 찾아낸 액션들에는 사용자의 항해 특성과 기호 등의 정보가 포함되어 있다. 하지만 본 논문에서 제안하는 것은 웹 애플리케이션 설계 시 항해적 측면에서 노드와 노드 사이를 효과적으로 연결하도록 하는 패턴을 찾아내는 것이다. 따라서 사용자의 개인적인 특성과 애플리케이션 고유의 설계 정보들을 여과할 필요가 있다. 따라서 해당 웹 애플리케이션 응용 도메인에 속하는 애플리케이션의 웹 서버 로그를 입력 값으로 이를 파싱하여 로우레벨의 액션을 찾아낸다. 액션에 대하여 웹 애플리케이션 안에서 의미 있는 액션을 골라내기 위하여 필터링 과정을 거친다. 필터링 과정에서는 응답 페이지가 웹 애플리케이션 내의 페이지가 아닌 것들과 웹 서버에서 발생하는 여러 페이지인 것들을 제거한다. 이렇게 필터링 된 액션 중에서 사용자가 수행하는 일련의 연속된 액션들을 찾아내도록 한다. 이것이 사용자가 수행하는 항해 경로가 된다. 이렇게 각각의 사용자가 수행한 항해 경로 정보는 정의 2에서와 같이 그래프로 표현이 된다. 하나의 웹 애플리케이션의 로그를 분석하여 만들어진 항해 경로 정보를 분석하여 많은 사용자에게 의해 반복되는 부분을 항해 패턴의 후보로 제안할 수 있을 것이다. 여러 애플리케이션에서 위와 같은 과정을 반복하면 각각의 애플리케이션에서 후보 패턴들을 산출해 내게 된다. 이 후보 패

턴들은 항해 경로 그래프의 일부로서 역시 노드와 간선들의 방향성을 가진 그래프로 나타나게 된다. 여러 후보 패턴들 중에서 반복되어 나타나는 부분 그래프를 항해 패턴으로 제안할 수 있을 것이다.

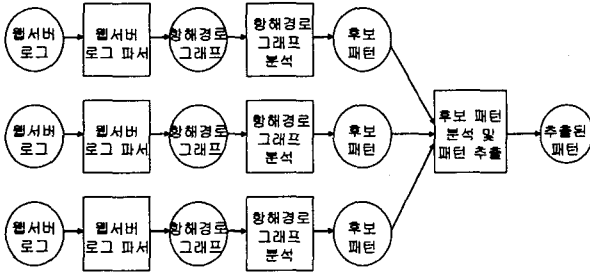


그림 3 패턴 추출 프로세스

4.2. 사례 연구

다음은 일반적인 도서관 시스템의 예이다. 사용자가 도서관 시스템에 접근했을 때 보여지는 첫 페이지  $p_1$ 를 노드  $n_1$ 이라고 한다. 사용자는 첫 페이지에서 한 번의 액션을 수행함으로써 다른 페이지  $p_2, p_3$ , 또는  $p_4$ 로 이동할 수 있다. 이 때 이 액션들은 각각  $e_1, e_2, e_3$ 의 간선으로 표현된다.  $p_2$ 에서는 하나의 액션  $e_4$ 를 수행하여  $p_3$ 로 접근이 가능하고,  $p_3$ 에서는 액션  $e_5$ 를 수행하여  $p_4$ 로 접근이 가능하다. 이것을 그래프로 표현하면 다음과 같다.

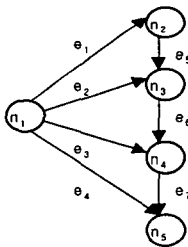


그림 4 도서관 시스템의 구조

사용자의 로그를 분석한 결과 ( $n_1, e_1, n_2, e_5, n_3$ ), ( $n_1, e_1, n_2, e_5, n_3, e_6, n_4$ )의 항해 경로가 발견되었다. 따라서 이 도서관 시스템에서는 ( $n_1, e_1, n_2, e_5, n_3$ )를 후보 항해 패턴으로 제안할 수 있다.

이와 같은 프로세스를 다른 웹 애플리케이션에 대해서도 수행하여 그 안에서 제안된 후보 패턴이 위의 것과 같다면 이것을 항해 패턴으로 제안할 수 있다.

5. 향후 계획

4장에서와 같은 프로세스를 거쳐 웹 애플리케이션의 액션들을 그룹화 할 때, 모든 애플리케이션에 적용할 수 있는 독립적인 기준을 찾아야 하겠다. 현재까지의 연구에서는 각 액션은 웹 애플리케이션의 종류에 의존하여 그룹화 되었다. 따라서 비슷한 성격을 갖는 애플리케이션을 분석하였을 때에 액션 유형을 찾아낼 수 있었다. 그렇지만, 도서관 시스템과 전자 상거래 시스템과 같이 서로 다른 목적을 갖는 웹 애플리케이션의 액션을 분석할 경우, 각 액션들이 갖는 성격을 일반화하는 방

법에 대한 연구가 필요하다.

6. 결론

본 논문에서는 웹 애플리케이션을 위한 설계 프레임워크에 관한 연구의 세부 과제로서 웹 애플리케이션의 항해 구조를 설계하는데 데 사용되는 웹 항해 패턴을 추출하기 위한 접근법을 기술하고 예를 들어 설명하였다. 웹 애플리케이션의 개발이 보편화되는데 반하여 아직까지 요구 사항, 설계 수준에서 이를 지원하기 위한 방법론의 개발은 미흡하다고 할 수 있다. 현재까지 UML을 확장하여 웹 애플리케이션을 모델링 할 수 있도록 하거나[1], 웹 애플리케이션을 네 가지 계층의 모델로 나누어 각각의 계층을 분리시켜 웹 애플리케이션의 모델을 고안할 수 있게 하는[2] 연구가 진행되었고, 항해 계층에서 적용 가능한 패턴을 제안한[6] 연구가 진행되었다. 그렇지만, 그러한 패턴을 추출해 내는 체계적 프로세스에 관한 논의는 부족하다. 본 연구를 통하여 웹 애플리케이션으로부터 패턴을 추출해 내기 위한 프로세스를 제안하였다. 로그를 분석한 결과 어느 노드와 어느 노드 사이의 항해가 어느 형태로 반복되는 지를 통하여 항해가 빈번히 일어나는 양상을 찾아내어 후보 패턴을 제안할 수 있었다. 패턴이 설계 시 생기는 문제에 대한 해법을 제시하고 있다는 점을 고려할 때, 후보 패턴이 되는 항해 경로 상의 각 노드들이 웹 애플리케이션에서 갖는 의미를 분석함으로써 패턴이 적용될 수 있는 상황에 대해 분석할 수 있을 것이다. 각 노드들이 웹 애플리케이션에서 갖는 의미를 고려할 경우, 문제의 복잡도가 매우 커지므로 이것을 효과적으로 추상화시킬 수 있는 방법에 대한 연구가 필요할 것이다.

7. 참고문헌

[1] D. Bonura, R. Culmone, and E. Merelli, "Patterns for Web Applications," *Proc. of Int' l Conf. on Software Eng. and Knowledge Eng.*, pp. 739-746, 2002.

[2] D. Schwabe and G. Rossi, "Developing Hypermedia Applications Using OOHD," *Proc. of Workshop on Hypermedia Development Process, Methods and Models*, Hypertext '98, 1998.

[3] G. Rossi, A. Garrido, and D. Schwabe, "Patterns Systems for Hypermedia," *Proc. of PLoP Conference*, Pattern Language of Program, 1997.

[4] A. Garrido, G. Rossi, and S. Carvalho, Design Patterns for Object-Oriented Hypermedia Applications, *Pattern Languages of Programs 2*. Vlissides, Coplien and Kerth eds., Addison Wesley, 1996.

[5] F. Lyardet, G. Rossi, and D. Schwabe, "Patterns for Adding Search Capabilities to Web Information Systems," *Proc. of EuroPloP*, 1999.

[6] G. Rossi, D. Schwabe, and F. Lyardet, "Patterns for Designing Navigable Information Spaces," *Pattern Languages of Programs IV*, Addison Wesley, 1999.

[7] J. Conallen, "Modeling Web Application Architectures with UML," *Communications of the ACM*, Vol. 42, No. 10, 1999.