

웹 응용 모델링의 요소 분류와 그에 따른 제약조건 기반 모델링

박영주⁰ 이기열 이병정* 김희천** 우치수

서울대학교 컴퓨터공학부
서울시립대학교 컴퓨터과학부*
한국방송통신대학교 컴퓨터과학부**

{ppang⁰, kylee, wuchisu}@selab.snu.ac.kr
bjlee@venus.uos.ac.kr*
hckim@knou.ac.kr**

Classification of Web Application Modeling Elements and Constraints Based Web Application Modeling

YoungJoo Park⁰, Keeyoull Lee, Byungjeong Lee, Heechern Kim, Chisu Wu
School of Computer Science and Engineering, Seoul National University
School of Computer Science, University of Seoul*
Dept. of Computer Science, Korea National Open University, Seoul, Korea**

요 약

웹 어플리케이션은 생명 주기가 짧고, 빠르고 정확한 개발이 요구되므로 모델링 단계에서의 요소 분류 및 요소간의 제약조건을 정의할 필요가 있다. 이에 본 연구에서는 각 프레임과 페이지를 그 성격에 따라 집합 프레임과 구조 프레임, 내용 페이지와 구조 정보 페이지로 분류, 이를 바탕으로 요소간의 관계를 정의하였다. 웹 응용 모델링의 요소 분류를 통해 그에 따른 제약 조건을 서술할 수 있는데, 이를 통해 사소한 예측할 수 있는 오류를 피할 수 있고, 웹 응용의 기본적인 테스트 케이스로도 사용할 수 있다.

1. 서론

초기의 웹 응용은 텍스트, 그래픽, 오디오, 비디오 등의 자료를 사용자에게 제시하기 위해 하이퍼미디어를 사용하는 것이 목적이었으므로, 그 자체는 매우 단순한 구조를 가졌다 [Pre01]. 또한 일시적인 사소한 오류가 발생하여도 제공자나 사용자에게 큰 피해가 없었으므로 어느 정도의 오류를 가지더라도 시장의 요구에 재빨리 대응 하는 것이 더 중요하게 여겨졌다. 하지만 점차 웹 응용의 구조가 복잡해지고, 응용 범위가 넓어짐에 따라 오류의 빈도도 높아지고, 오류가 발생했을 경우의 피해도 심각해지고 있다. 굳이 금융권의 예를 들지 않더라도 반복되는 사소한 오류는 사용자 기피의 원인이 될 수 있다.

이에 본 논문은 제약조건 기반 웹 응용 모델링을 제시한다. 웹 응용의 모델링은 공간개념인 프레임과 실제 사용자에게 제공되는 정보인 페이지로 이루어진다. 본 연구에서는 각 프레임과 페이지를 그 성격에 따라 분류 하고, 이를 바탕으로 요소간의 관계를 정의하였다. 웹 응용 모델링의 요소 분류와 그에 따른 제약 조건을 서술하는 것으로 사소한 예측할 수 있는 오류를 피할 수 있고, 이는 테스트의 요소로도 사용할 수 있다.

본 논문의 구성은 다음과 같다.

2장에서 관련 연구들을 살펴본다. 3장에서는 웹 어플리케이션의 구성 요소들을 정의하고, 4장에서 웹 어플리케이션이 가져야 할 디자인 단계에서의 제한 사항 등을 정의하였다. 마지막 5장에서는 본 연구에서 수행하지 못한 요건들을 살펴보고 결론을 내린다.

2. 관련연구

[Con03]은 웹 응용의 모델링을 위한 UML의 확장을 통해 많은 새로운 개념들을 도입하여 프레임 집합, 프레임과 페이지의 관계 등을 정의하고 있으나, 프레임의 구성이나 페이지의 내용에 따른 분류, 그들 간의 관계를 정의하지 않고 있다.

[Pie03]의 경우 webfile과 zone의 개념을 도입하여 프레임과 페이지를 분리하고 그 관계를 설명하고 있으나 역시 그들 사이의 제약에 관한 정의를 하고 있지 않다.

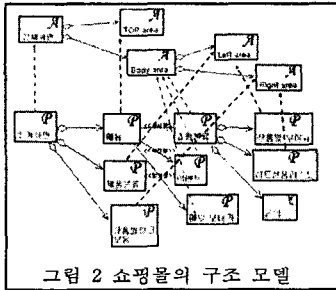
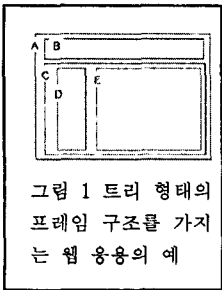
[Gil00]에서는 각 페이지들을 루트 페이지와 중간(intermediate) 페이지, 말단 페이지로 분류하고 있으나, 이는 다양한 향해 패턴을 설명하기 위한 내용을 담고 있는 페이지들의 분류만을 다루고 있다.

3. 웹 어플리케이션의 분석을 위한 디자인 단계에서의 정의

웹 어플리케이션은 크게 페이지와 프레임을 구성 요소로 가진다. 프레임은 일반적으로 트리 구조로 이루어져 있다. 각각의

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

프레임은 하나의 말단 프레임이거나 혹은 복수개의 프레임을 포함하는 집합프레임으로 분류해 볼 수 있다. 웹 어플리케이션 초기에 프레임은 단지 공간의 분할 개념을 포함하여 프레임 간의 관계만을 가졌으나 웹 어플리케이션에서의 다양하고 효율적인 표현을 위하여 고안된 iframe의 경우 페이지 안에 프레임이 삽입되기도 한다. 페이지는 사용자에게 보여줄 내용을 포함하거나 서버에서 연산이 되는 단위이다. 페이지 역시 초기에는 주로 정적인 페이지들로 구성되어 정보를 제공하는 것에 초점을 맞추었으나, 웹의 이용 범위가 확대되면서 사용자와의 상호 교류를 중시하게 되었다. 이에 입력 폼으로부터 사용자의 입력을 받아 동적으로 생성되는 페이지의 중요성이 증가하게 되었다.



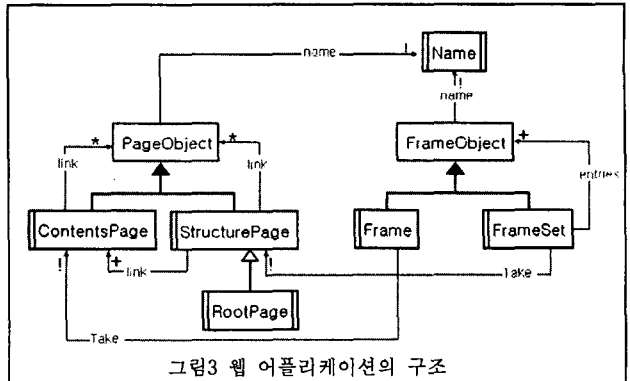
프레임은 그 기능에 따라 두 종류로 나누어 볼 수 있다. 하나의 공간인 프레임과 프레임들의 집합인 집합프레임(frameset)이다. 그림 1에서 A와 C는 집합 프레임으로 A는 내부에 프레임 B와 집합 프레임 C를 포함하고, C는 복수개의 프레임을 포함한다. B와 D, E는 프레임으로 각각 그 공간에 웹 응용의 실제 내용이 나타난다. 프레임의 구조를 펼쳐보면 그림 2의 위쪽과 같은 트리 형태로 트리의 말단 노드에는 프레임이 위치하고, 자식 노드를 가지는 노드는 집합 프레임이 된다 [Par03]. 페이지는 생성 시기에 따라 사용자의 입력을 받아 서버에서 동적으로 생성되는 페이지와 정적인 페이지로 나눌 수 있다. 또한 포함하는 내용에 따라 내용만을 가지는 페이지, iframe 과 같은 구조 정보와 내용을 함께 가지는 페이지, 프레임 구조 정보만을 가지는 페이지로 분류해 볼 수 있다. 그림 2의 아래쪽 트리가 페이지들의 관계를 나타내는 것으로 다른 페이지들을 포함하는 '초기화면', '쇼핑분류', '이벤트' 등은 집합 프레임에 사상되는 구조 정보를 가지는 페이지이고, '메뉴', '제품분류', '상품별 부메뉴' 등은 프레임에 사상되는 내용 정보를 가지는 페이지이다. iframe을 가지는 페이지는 내용을 가지는 페이지와 구조 정보를 가지는 페이지의 속성을 동시에 지니고 있다고 말할 수 있으므로 본 연구에서는 사용자에게 제공할 내용을 가지는 페이지와 구조정보를 가지는 페이지, 이렇게 2가지로 구분하도록 한다.

기본적으로 웹 응용에서 사용자는 링크를 통해 항해를 하게 된다. 링크는 일반적으로 항해의 도착 페이지와 해당 페이지가 나타날 프레임 정보를 포함한다. 지정된 페이지가 정적인 페이지인 경우 해당 페이지의 내용을, 동적인 페이지의 경우 서버

에서 수행된 연산 결과를 지정한 프레임에 보여준다. 프레임이 지정되어 있지 않은 경우에는 현재 작업 중인 프레임에 새로운 페이지가 갱신된다. 단순한 하이퍼링크 이외의 submit, redirect의 경우 역시 그와 유사한 성격을 지니므로 본 논문에서는 링크만을 고려하도록 한다.

4. 웹 어플리케이션 모델링

이상의 웹 어플리케이션의 정의를 바탕으로 웹 어플리케이션을 모델링 하면 다음과 같이 표현 할 수 있다.



1. 페이지 객체는 컨텐츠페이지 집합과 구조페이지 집합으로 분할(partition)되어 있다. 구조 페이지에는 유일한 루트페이지가 존재한다.
2. 프레임객체는 프레임 집합과 집합 프레임의 집합으로 분할되어 있다.
3. 각각의 페이지객체와 프레임객체는 서로 구별되는 이름을 가진다.
4. 하나의 프레임은 단 하나의 페이지와 take 관계를 가진다. 즉, 하나의 프레임에 동시에 두 개의 페이지가 동시에 보일 수 없다. 또한, 프레임에 할당되어 사용자에게 보이는 페이지가 존재하지 않는 경우는 없어야 한다. 단, 프레임에는 컨텐츠 페이지가 할당되어야 하고, 집합프레임에는 구조페이지가 할당되어야 한다.
5. 컨텐츠 페이지와 구조페이지는 각각 link의 관계를 가진다. 모든 페이지는 0개 이상의 페이지와 링크 관계를 가져야 한다. 또한, 구조페이지는 반드시 하나 이상의 내용페이지와 링크 관계가 있어야 한다.
6. 모든 페이지는 루트페이지로부터 여러 번의 link를 통해 모두 도달 가능해야 한다.

위의 정의를 바탕으로 모델링을 위한 제약 조건들을 Alloy로 표현하면 다음과 같다.

```

1 module WebApplication
2
3 sig PageObj {link : set PageObj, name: one NameObj}
    
```

```

4 sig ContentPage extends PageObj {
5 sig StructPage extends PageObj {
6 one sig RootPage extends StructPage {
7 sig FrameObj (name : one NameObj)
8 sig Frame extends FrameObj {take: one ContentPage}
9
10 sig FrameSet extends FrameObj {take: one
    StructPage, entry: some FrameObj}
11 sig NameObj {
12
13 fact UniqueName { (all p1, p2 :PageObj | p1.name =
    p2.name => p1=p2)
14     && (all f1, f2 :FrameObj | f1.name = f2.name
    => f1 = f2)
15     && (all p: PageObj | no f:FrameObj | p.name =
    f.name)
16     && (all f:FrameObj | no p: PageObj | p.name =
    f.name) }
17
18 assert TakeOnePage { all s:StructPage | some c:
    ContentPage | c = s.link }
19
20 assert LinkIsNotOwn { all p:ContentPage | p ! in
    p.link }
21 check LinkIsNotOwn for 8
22
23 pred Reachable () { PageObj in RootPage.*link }
24 run Reachable for 8
    
```

그림 4 안전하게 동작하는 웹 응용의 제약 조건

- 3 : PageObj는 하나의 NameObj 과 name 관계를, PageObj집합과 링크 관계를 가진다. 각각의 페이지는 항해를 수행할 때 이름으로 호출되므로 모든 페이지는 고유한 구별되는 이름을 가져야 한다. 또한 페이지는 0 혹은 그 이상의 페이지와 링크 관계를 가져야만 항해가 이루어 질 수 있다.
- 4, 5 : ContentPage와 StructPage는 PageObj를 상속받는다.
- 6 : RootPage는 StructPage에 속하는데, 단 하나가 유일하게 존재한다. RootPage는 해당 웹 어플리케이션을 실행하였을 때 제일 먼저 실행되는 페이지로써 본 연구에서는 다수의 프레임 을 갖는 웹 응용을 가정 하였으므로 RootPage는 StructurePage이다.
- 7 : FrameObj는 고유한 이름을 가진다. 프레임 역시 서로 이름으로 구별되므로 같은 이름을 가지는 프레임이 존재해서는 안된다.
- 8 : Frame 은 FrameObj 를 상속받는데, 오직 하나의 ContentPage와 take 관계를 가진다. 프레임이 페이지와 take 관계라는 것은 해당 페이지가 화면에 출력되어 사용자에게 보

여진다는 의미이므로 한 프레임이 두 개의 페이지와 take관계를 가질 수 없다.

- 10 : FrameSet 은 FrameObj 를 상속한다. 하나의 StructPage와 take 관계를 가지고, 복수개의 프레임과 entry 관계를 가진다.
- 13~16 : 모든 PageObj와 FrameObj는 서로 구별되는 이름을 가져야 한다.
- 18 : Struct Page는 적어도 하나 이상의 Content page와 link 관계를 가져야 한다.
- 20 : Content page는 자기자신과 link 관계를 가질 수 없다.
- 23 : 모든 page는 RootPage로부터 여러 번의 링크 관계를 따라 접근이 가능해야 한다.

7. 결론

웹 어플리케이션의 동작이 진행된다는 것은 새로운 페이지들이 생성되고, 각 페이지의 기능이 수행되는 것을 의미한다. 페이지가 가지는 기능은 내용을 보여 주고, 새로운 프레임의 형태를 만들고, 입력을 바탕으로 서버에서 연산을 수행하고, 링크를 따라 항해를 하는 것이라고 말할 수 있다. 따라서 웹 어플리케이션이 안전하게 수행된다는 것은 프레임에 제대로 페이지가 할당되어 사용자가 볼 수 있고, 항해를 여러 번 수행한 후에도 프레임과 페이지간의 할당에 문제가 발생하지 않는다는 것을 의미한다. 이를 위해 웹 어플리케이션의 개체들과 기능을 정의 하고 몇 가지 제약 조건을 정의하였다. 요소 분류와 제약 조건 정의를 통해 사소한 오류의 방지 및 테스트 케이스 생성이 쉬워진다. [Par04]에서 정의한 다양한 항해 방식에 따른 제약 조건 정의의 추가가 수행되어야 하겠다.

8. 참고문헌

- [Pre01] R. S. Pressman, *Software Engineering : a Practitioner's Approach* 5th ed., McGraw Hill, 2001
- [Con03] J. Conallen, *Building Web Applications with UML*, Addison Wesley, 2003.
- [Par03] 박영주, 이기열, 신우창, 이병정, 김희천, 우치수, "확장형 UML을 이용한 웹 애플리케이션 모델링 기법," 정보과학회 추계 학술발표논문집(2), 제30권, 제2호, October 2003.
- [Par04] 박영주, 이기열, 이병정, 김희천, 우치수, "확장형 UML을 이용한 웹 애플리케이션 모델링을 위한 항해의 분류", 정보과학회 춘계 학술발표논문집(B), 제31권, pp. 397-399, Apr. 2004
- [Pie03] Pierre-Alain Muller, Philippe Studer, Jean Beivin: Platform Independent Web Applikation Modeling, Sixth International Conference on the Unified Modeling Language - the Language and its applications, 2003
- [Gil00] Mark L. Gillenson, Daniel L.Sherrel and lei-da Chen, "A Taxonomy of Web Site Traversal Patterns and Structures," *Communications of AIS*, Vol.3, No.17, 2000.