한국컴퓨터종합학술대회 2005 논문집 Vol. 32, No. 1(B)

UML 2.0 행위 다이어그램을 확장한 웹 응용의 향해 모델

박상현* 이욱진 이병정** 김희찬** 우치수
서울대학교 컴퓨터공학부, 서울시립대학교 컴퓨터과학부*, 한국방송통신대학교 컴퓨터과학과**
{zez4shy*, duri96, wuchisul}@selaab.snu.ac.kr, bljee@venus.uos.ac.kr*, hckim@knu.ac.kr**

A Navigation Model of Web Applications with Extended Behavioral Diagrams of UML 2.0

Sanghyun Park*, Wook-jin Lee ByungJeong Lee*, Heechern Kim**, Chisu Wu
School of Computer Science & Engineering, Seoul National University*
School of Computer Science, University of Seoul*
Dept. of Computer Science, Korea National Open University**

요 약

항해하는 웹 응용의 대표적인 행위 특성들이 본 연구에서는 UML 2.0의 행위 다이어그램에 대해서 모델링을 확장한 행위 응용을 제안한다. 본 행위 모델은 뷰 관점 행해 모델과 데이터 전송 관점 행해 모델로 구성된다. 뷰 관점 행해 모델은 UML 상태 기계 다이어그램을 확장하여 사용자에게 보이고 있는 환경을 기술한다. 데이터 전송 관점 행해 모델은 데이터가 전송되는 행해를 나타내며 UML 시퀀스 다이어그램을 확장하여 표현한다. 두 환해 모델은 상호 보완적으로 사용하여 관련된 환해 를 훈련한다. 본 논문에서는 UML 2.0의 행위 모델의 확장점과 향해 모델의 표기법을 제시하고, 사례 연구를 통하여 실제적인 향해 모델의 예를 보인다.

1. 서론

항해(navigation)는 웹 응용(web applications)에 중요한 행위 특성이 행해의 표현 계층에 국한되며 기술되고 있다. 특히 행해 응용의 회차는 다양한 비즈니스 논리(business logic)를 포함하게 되며 따라 환해 계층은 비즈니스 논리 계층까지 확장되었다. 따라서 행해 구조를 응용하게 모델링하기 위해서는 클라이언트 및 서버 페이지로 분할 비즈니스 논리 구현한 컨포넌트까지 고려할 필요가 있다.

웹 응용의 항해 구조를 모델링하기 위해 여러 가지 연구가 진행되었다. [1, 2, 3]에서 행해 응용의 구성 요소의 그룹 간의 관계를 구성하는 개념 모델(Conceptual Model)을 만들고, 여기에서 행해 모델을 유도한다. 그러나 이를 행해 모델은 표준 계층에서의 항해에 비해 비즈니스 논리를 포함하는 항해에 대하여 고려가 부족하다.[2, 4] 등은 UML 클래식 다이어그램(Class Diagram)을 확장한 행해 모델을 제안하였다. 그런데 UML 클래식 다이어그램은 구조적 모델을 위한 다이어그램으로 항해의 형제 특성의 부재를 반영하는 한계가 있다.


논문의 구성은 다음과 같다. 1장에서는 본 연구에서 제안하는 항해 모델링, 2장에서는 이를 적용한 모델링 사례를 보인다. 3장에서는 연구의 결론을 도출하고 향후 연구 과제를 제시한다.

2. 항해 모델

본 장에서는 행해 응용의 구성 요소를 우선 식별하고, 항해의 형제 특성을 보이나. 또 항해를 뷰 관점 행해 모델과 데이터 전송 관점 행해 모델로 표현한다.

2.1 행해 응용의 구성

행해 응용은 크게 웹 페이지, 자원, 컨포넌트로 구성된다[6]. 웹 페이지는 클라이언트(client) 페이지와 서버(server) 페이지로 분류된다. 클라이언트 페이지는 HTML 문서와 같이 클라이언트 컴퓨터로 보내어 보이지 않는 페이지로 정적 페이지다. 서버 페이지에는 JSP, ASP, PHP 등이 있으며 동적 특성을 지닌다. 컨포넌트는 또한 실행 환경에 따라 클라이언트 및 서버 컨포넌트로 분류된다. 서버 컨포넌트는 서버 페이지에 의해 호출되어 특정 기능을 수행하는 컨포넌트로 EJB, NET 응용 컨포넌트, 웹 서비스(Web Services)등이 있다. 클라이언트 컨포넌트는 ActiveX, OCA, Java Applet, Flash 등 클라이언트 단계에서 작동하는 컨포넌트이다.

웹 페이지의 분야에 따라 행해 응용의 항해는 클라이언트 페이지에서 클라이언트 페이지로의 항해(서버 컨포넌트), 클라이언트 페이지에서 서버 페이지로의 항해(서버 컨포넌트), 서버 페이지에서 서버 페이지로의 항해(서버 컨포넌트)의 순으로 진행된다. 본 연구에서는 일반적으로 클라이언트 페이지에서는 행해 응용의 표현 계층에 서버 페이지와 서버 컨포넌트가 비즈니스 논리 계층에 속한다. CON은 일반적으로 웹 페이지 간에 전달되는 데이터가 없는 단순 응용이다. 반면 서버로 제목의 환경 또한 페이지의 두 가지 전달방식이 존재한다. 예를 들어 로그인 상태에서 접속할 수 있는 페이지의 경우 클라이언트 페이지는 서버의 서비스(서버 컨포넌트)로 전달되어서 접속할 수 있다. 뷰를 통해 서버 컨포넌트를 계층화하는 사례는 효과적이다.
서버 페이지에서 클라이언트 페이지로 로그인 정보가 전달되어 오므로 데이터의 전송이 이루어진다.
본 연구에서는 항해를 뷰 관리 및 데이터 전송 과정에 따라


graham.1. 상대 기계 다이어그램의 정의


표 1. 확장된 노드(node) 및 경로(path) 타임의 표기법

<table>
<thead>
<tr>
<th>노드(node) 타입</th>
<th>표기</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageView</td>
<td></td>
</tr>
<tr>
<td>VirtualPageView</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>경로(path) 타입</th>
<th>표기</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation</td>
<td></td>
</tr>
<tr>
<td>BizNavigation</td>
<td></td>
</tr>
</tbody>
</table>


표 1은 확장된 UML 상대 기계 다이어그램의 노드(node) 타임 및 경로(path) 타임의 표기법을 보여준다. 표 2는 대표적인 항해 유형별 뷰 관점 항해 모델로의 매핑을 나타낸다. 뷰 관점 항해 모델에서 모든 항해의 결과는 클라이언트 페이지로 귀결된다는 점을 나타낸다. 이는 웹 응용이 사용자와의 상호작용을 바탕으로 만들어진 속성을 나타나기 때문이다. 따라서 CSN은 SCN에 연결되어야 한다.

표 2. 대표적 항해 유형에 따른 표기법

<table>
<thead>
<tr>
<th>항해유형</th>
<th>의미</th>
<th>표기</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCN</td>
<td>클라이언트 페이지 'C1'에서 클라이언트 페이지 'C2'로 항해.</td>
<td>C1(C2)</td>
</tr>
<tr>
<td>CSN/SCN</td>
<td>클라이언트 페이지 'C1'에서 서버 페이지 'C2'로 항해 후 다시 클라이언트 페이지 'C1'으로 항해. 서버 페이지 'C2'로 정보를 전달하고 클라이언트 페이지로 항해, S1로 항하도록 행동.</td>
<td>C1(C2)</td>
</tr>
</tbody>
</table>

다음은 상대 기계 다이어그램의 주요 요소와 그에 따른 뷰 관점 항해 모델에서의 의미를 기술한 것이다.

- 지역(Region): 페이지가 여려 페이지로 복합된 경우 프레임 구성 등 각각의 페이지를 표현하기 위하여 페이지 노드를 분할
- 시작 의의 상태(initial pseudo state): 지역으로 분할되어 부 다이어그램(sub diagram)으로 표현된 각 페이지의 시작 상태를 등기화
- 트리거(trigger) 및 가드(guard): 항해의 이어지는 조건 및 이벤트 표현(예: 버튼 클릭)

2.3 데이터 전송 관점 항해 모델

데이터 전송 관점 항해 모델은 CSN, SCN, SSN 등 데이터가 전송되는 항해를 기술하는데 쓰인다. 본 모델은 웹 응용의 표현 계층과 비즈니스 논리 계층 간의 연결 지점을 보유한 컴퓨터 네트워크 구현을 위한 맥락이 된다.

그림 2. 시스템 다이어그램의 확장

데이터 전송 관점 항해 모델은 UML 2.0의 시스템 다이어그램을 확장하여 표현한다. 시스템 다이어그램은 단일한 웹에 게이블 속속적이며, 따라서 웹 응용의 표현 계층과 비즈니스 논리 계층과 연결되는 지점별로 모델을 작성한다. 각각의 모델이 만들어진 후 선행된 컴퓨터의 기능에 따라 컴퓨터로 그룹화 및 통합이 이루어질 수 있다. 그림 2는 시스템 다이어그램 메타 모델의 확장된 부분을 나타낸다. LineLife 클래스는 클라이언트 페이지, 서버 페이지 및 컴퓨터를 위한 LineLife 클래스로 확장하였다. LineLife 클래스는 BasicInteractions 페이저에
속한다. 표 3은 확정된 노드의 표기법이다.

### 표 3. 확정된 노드의 표기법

<table>
<thead>
<tr>
<th>노드(none)</th>
<th>표기</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clientpage/lifetime</td>
<td>cpName</td>
</tr>
<tr>
<td>Serverpage/lifetime</td>
<td>spName</td>
</tr>
<tr>
<td>Component/lifetime</td>
<td>cmpName</td>
</tr>
</tbody>
</table>

이외의 데이터 전송 환경 모델 요소는 서브스 디자이그램의 요소 정의를 따른다.

3. 항해 모델링 사례 연구


![index.html](http://example.com/index.html)

**그림 3. 뷰 관점 항해 모델의 일부(PaGeS)**

그림 3은 뷰 관점 항해 모델의 일부로 뷰 전송을 개발 관점에서 얻어낸 것임을 보여준다. index.html의 뷰 전송 구조로 볼 때, 폰탈 뷰기지대가 만들어져 있으며, 이 뷰기지대는 3개의 파일로 나뉘며, index.html 폰탈 뷰기지대의 3개 파일의 각 뷰 파일은 각 뷰 파일에서의 뷰 파일(sub)을 포함한다. 항해는 자신 혹은 다른 뷰기지대의 뷰 파일에 의해 이루어진다. b_search.html과 a_search.html의 뷰 파일은 같은 Search.class 클래스에서 사용하고, 검색 결과는 searchResult.jsp에 의해 포트폴리오로 제공된다.

4. 결론과 향후 연구

본 연구에서는 웹 응용의 항해를 모델링하기 위한 뷰 관점 및 데이터 전송 관점 항해 모델을 제안하였다. 두 항해 모델은 각각 UML 2.0의 상태 기계 디자이그램과 서브스 디자이그램과의 모든 구조를 확장하여 표현하였다. 이들은 상호보완적으로 작동하여 웹 응용의 항해 모델을 제공한다. UML 통행 디자이그램은 전송 항해 모델을 확장하였으며, 본 항해 모델은 통행 모의 자원문 제로써, 또한 UML의 기본적 성질인 가독성과 정형성을 다루는 데 도움이 있다.

현재의 뷰 관점 항해 모델은 항해의 수치 증가에 따라 복잡도가 급격히 늘어나는 문제로 있다. 이러한 문제를 해결하기 위한 항해 모델의 확장 및 단순화(slicing) 방안이 필요하다. 데이터 전송 관점 항해 모델의 경우 품목별 셀과 구현 구조와 직접 연관될 수 있도록 유사한 항해 구조를 적용하는 방안에 대해 연구 중이다.

이 외에도 항해 모델링을 지원하는 도구와 기존의 레거시 웹 응용의 항해 모델을 확장하여 항해 모델을 구현해내는 프로세스에 대한 연구를 진행할 예정이다.

**Acknowledgement**

본 연구는 한국학회대회 특기정상연구(R01-2002-000-00135-6) 지원으로 수행되었다.

**참고 문헌**


