

웹 애플리케이션 프리젠테이션 계층 프레임워크 생성을 위한 XML 기반 문서화

김송주⁰ 유철중 장옥배
전북대학교 컴퓨터과학과

sjkim@se.chonbuk.ac.kr
{cjyoo, okjang}@moak.chonbuk.ac.kr

XML-based Documentation for Generating Web Application Presentaion Layer Framework

Song-Ju Kim⁰ Cheol-Jung Yoo Ok-Bae Chang
Dept. of computer Science, Chonbuk National University

요 약

J2EE 기반의 웹 애플리케이션의 개발의 장점은 개발자로 하여금 시스템적 문제들에 대한 부담을 절감시켜 비즈니스 로직 개발에 중점을 두도록 하는데 있다. 그러나 시스템적 문제들에 대한 부담은 많이 사라진 반면 실질적인 비즈니스 로직을 개발하는 개발자가 프리젠테이션 계층의 개발 부분에도 참여해야 하고 프리젠테이션 계층에서 사용되는 여러 기술들도 익혀야 하는 부담을 가지게 된다. 따라서 본 연구에서는 이러한 부담을 최소화하고, 웹 애플리케이션 프리젠테이션 계층 프레임워크를 신뢰성 있게 구축하며 프레임워크를 자동 생성하고자 프레임워크를 명세하여 XML 기반의 문서로 변환한다. 이렇게 함으로써 웹 애플리케이션 서버에 의존적인 것들과 또 다른 많은 것들을 확장할 수 있는 여지를 남겨둠으로써 사용자의 편의성을 증가시킬 수 있다.

1. 서 론

J2EE 기반의 웹 애플리케이션에서 제시하고 있는 비전은 기존에 개발자들이 부담해야 했던 트랜잭션, 동시성제어, 보안 처리등의 시스템적인 것들을 EJB 컨테이너가 담당하게 함으로써 개발자들이 비즈니스 로직에만 전념할 수 있도록 하겠다는 것이었다[1]. 그러나 실제 웹 애플리케이션 개발을 고려해 볼 때 EJB 컴포넌트를 비즈니스 컴포넌트로 활용하여 웹 애플리케이션을 구축하는 것은 개발자에게 기존의 시스템적인 것들을 제외시켰지만 오히려 프리젠테이션 계층 설계에 계속 참여하도록 하여 부담을 증가시킨 것으로 알려져 있다. 그 이유로는 기본적인 MVC 모델을 고려해 볼 때 프리젠테이션 계층은 여러 가지 정보를 비즈니스 계층에 보내고 이러한 정보에 기반하여 비즈니스 계층에서 넘어오는 내용을 프리젠테이션 계층이 표현하기 때문이다. 즉, MVC 구조를 고려해 보면 서로 역할의 분담이 나누어져 있기는 하지만 서로 유기적으로 협동함을 알 수 있다. 따라서 개발자에게 미리 웹 애플리케이션의 프리젠테이션 계층을 프레임워크화하

여 전달하여 준다면 좀더 빠르고 유연하게 비즈니스 계층 개발자가 웹 애플리케이션을 작성할 수 있다.

본 논문에서는 비즈니스 계층 개발자에게도 표준이 되고 프리젠테이션 계층의 개발자에게도 표준이 될 수 있는 웹 애플리케이션의 프리젠테이션 계층 프레임워크를 XML 기반으로 문서화 한다. 문서화의 근간은 우선 먼저 OCL로 웹 애플리케이션의 프리젠테이션 레이어를 명세하는 것이다. 이러한 명세의 목표는 기본적인 웹 애플리케이션 프리젠테이션 계층의 역할과 웹 컴포넌트간의 협동자들을 DBC 접근법을 통해 각 컴포넌트 사이의 책임 및 의무를 정확히 명세하고 명세된 내용과 프레임워크의 프로즌 스팟(Frozen Spot)부분을 XML 기반으로 문서화하여 이를 기반으로 웹 애플리케이션 프리젠테이션 계층의 자동 생성을 지원하기 위함이다.

더 나아가 프로즌 스팟 부분을 XML로 문서화하고 경우에 따라 핫 스팟 부분도 문서화할 수 있도록 유연성을 부여함으로써 새로운 프로즌 스팟과 핫 스팟(Hot Spot)을 추가할 수 있도록 하여 프리젠테이션 계층 개발자나 비즈니스 계층 개발자가 손쉽게 프레임워크를 커스터마이징 할 수 있도록

하였다.

본 논문의 구성을 보면 먼저 2장에서 웹 애플리케이션 프리젠테이션 계층 프레임워크에 대해서 설명하고 3장에서는 프레임워크를 XML로 문서화 하기 이전에 프레임워크의 구성요소간의 책임과 의무를 명확히 하기 위해 명세언어를 이용하여 명세하고, 4장에서는 명세된 프레임워크를 XML로 문서화한 예를 보였다. 마지막으로 5장에서는 본 연구의 최종 목표인 웹 애플리케이션 프리젠테이션 계층의 소스 코드 자동 생성이 가지는 의미와 향후 연구 방향을 제시한다.

2. 관련연구

2.1 웹 애플리케이션 프리젠테이션 계층 프레임워크

본 연구에서 예로 사용하는 프리젠테이션 계층 프레임워크는 자체 개발한 프레임워크를 사용하였다. 본 프레임워크는 기존의 프레임워크와 같이 구조적으로는 MVC 패턴을 따르고 있으며 클라이언트가 웹 브라우저라는 가정 하에 설계되었다[2].

프레임워크는 웹 티어가 가져야 할 역할들 중 프리젠테이션 계층만을 대상으로 하였고 각 티어를 가로질러 적용 될 수 있는 이벤트 관리자[3]와 비즈니스 로직 처리자 선택기는 프리젠테이션 계층에서 해야 할 일만을 정의하였고 MVC 패턴에서 모델에 해당하는 나머지 부분과 서로 통신할 수 있도록 표준 인터페이스도 정의하였다. 프레임워크의 구조도는 (그림 1)과 같다.

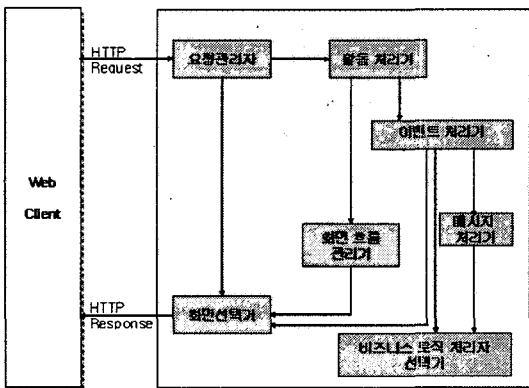


그림 1. 웹 애플리케이션 프리젠테이션 프레임워크

2.2 DBC 접근법을 기초로 하는 프레임워크의 명세

DBC(Design By Contract) 접근법은 다음과 같은 이점을 지님으로써 소프트웨어의 신뢰성을 높일 수 있는 명세 기법이다[4].

- ◎ 애플리케이션 로직으로부터 제약사항 검토를 분리시킴으로써 테스팅에 소모되는 노력을 감소시킨다.
- ◎ 실패의 발생에서 결함으로 이어지는 상황을 쉽게 확인할 수 있도록 가시성(observability)을 개선시켜줌으로써 디버깅에 드는 노력을 감소시킨다.
- ◎ 인터페이스에 대한 가장 최신 정보를 유지하고 모호성이 없는 문서를 제공할 수 있도록 해 준다.

이러한 이점 이외에도 다른 많은 이점들을 여러 문헌에서 찾아 볼 수 있다. 예를 들면 ISE (Interactive Software Engineering)에서는 다음과 같이 기술하였다[5].

- ◎ 객체지향 방법론뿐만 아니라 더 나아가 소프트웨어 구축에 대한 이해력을 높여준다.
- ◎ 버그 없는 객체지향 시스템 구축을 위한 체계적인 접근법이다.
- ◎ 디버깅, 테스팅, 더 나아가 품질 보증을 위한 효과적인 프레임워크이다.
- ◎ 소프트웨어 컴포넌트의 문서화를 위한 훌륭한 방법론이다.
- ◎ 비정상적인 상황을 다루기 위한 기법으로 예외처리를 위한 안전하고 효과적인 방법론이다.

이러한 이점들을 살펴볼 때 DBC접근법은 복잡하고 디버깅과 테스팅 그리고 품질보증을 하기 어려워지는 웹 애플리케이션의 개발에 알맞은 접근법이라고 볼 수 있다.

따라서 본 논문에서는 프레임워크의 책임 및 권한을 DBC 접근법을 이용하여 명세하였다. 이러한 접근법은 사전조건, 사후조건, 불변조건으로 소프트

웨어를 명세하는 접근법을 말하며 이 접근법을 이용하는 명세 언어 중 본 논문에서 채택한 OCL 명세 언어의 예는 (그림 2)과 같다.

```

context: VideoAccount::release(aVideo:Video)
Pre:
    store.stock->count(aVideo) = 1
    aVideo.available
    self.hired->size < self.videoHireLimit
    self.hired->count(aVideo) = 0
Post:
    store.stock->count(aVideo) = 1
    aVideo.hired
    store.stock->size = store.stock@pre
    self.hired->count(aVideo) = 1
    self.hired->size =self.hired@pre + 1
    
```

그림 2. OCL 명세언어의 예

3. 프레임워크의 소스코드 자동 생성을 위한 XML 문서화

DBC 접근법을 적용한 프레임워크의 OCL 명세에는 실제 애플리케이션 로직과 관련된 소스 코드를 포함하지 않는다. 따라서 본 논문에서 궁극적으로 추구하고자 하는 웹 애플리케이션 프리젠테이션 계층 프레임워크 소스 코드 생성을 위해 OCL로 명세된 내용을 XML로 문서화시키면서 실제 코드 부분은 XML태그로 정의하여 포함시켰다[6]. 이러한 문서화 과정을 보여주기 위해 활동 처리기의 XML 문서화 예 (그림 3)에 나타내었다.

이러한 활동처리기는 일반적인 컨트롤러의 역할을 담당하는 부분으로 일반적인 MVC 컨트롤러의 역할을 담당하고 있다 또한 웹 컴포넌트의 메시지 전달 비동기성 지원을 위해 새로운 메커니즘을 포함하는 메시지 처리기를 추가하였고 MVC 모델을 웹 티어에 둘 것인지 아니면 EJB 티어에 둘 것인지에 대한 선택을 할 수 있는 모델 위치 선택기가 존재한다. 이러한 선택기는 웹 티어와 EJB 티어 모두에 MVC 모델 부분을 둘 수 있는 것 또한 지원하고 있다. (그림 3)은 활동 처리기의 프론트(Front)부분의 예를 보여준다.

```

<?xml version = "1.0">
<component name = "Main">
<role> controller </role>
<comment>웹 화면에서 오는 모든 요청을 받아들이는 프론트(Front) 컨트롤러 </comment>
...
<structure>
<classes>
<class name = "DoProcessor" scope="public">
<conditions>
<pre>none</pre>
<post>
.....
</post>
<invariant>
.....
</invariant>
</conditions>
<methods>
<method name="doProcess" scope = "public"
arguments="HttpServletRequest,request" />
<code = "Servlet">
    Event ev = null;
    .....
    EVENTAction action =
        getAction(selectedURL);
    if (action != null) {
        action.setServletContext(context);
        action.doStart(request);
        ev = action.perform(request);
        .....
        action.doEnd(request, eventResponse);
    }
</code>
</methods>
<attributes>
<attribute name="aciton" scope = "public"
type="EVENTAction" />
.....
</attributes>
</classes>
</component>
    
```

그림 3. 활동처리기 컴포넌트의 XML 문서화의 예

본 문서의 구조는 OCL 명세의 구조를 적절한 가공을 통해 XML 태그를 이용해 표현하였으며 이에 덧붙여 <codes> 태그를 통해 웹 애플리케이션의 코드를 표현하였다. 각 태그의 의미는 (표 1) 과 같다.

참고문헌

표 1. 각 태그의 의미

태그	의미
<component>	웹 컴포넌트의 이름
<role>	컴포넌트의 역할
<comment>	컴포넌트에 대한 설명
<structure>	컴포넌트의 전체 구조를 표현
<classes>	컴포넌트를 이루는 클래스 집합을 의미
<class>	각 단위 클래스들을 기술
<conditions>	클래스가 갖추어야 하는 조건 기술
<pre>	OCL의 사전조건 기술
<post>	OCL의 사후조건 기술
<invariant>	OCL의 불변조건 기술
<methods>	클래스의 메소드들의 집합을 의미
<method>	각 단위 메소드들을 기술
<code>	단위 메소드의 코드를 기술
<attributes>	컴포넌트의 속성들의 집합을 의미
<attribute>	각 단위 속성들을 기술

4. 결론 및 향후 연구

본 논문에서는 웹 애플리케이션 프리젠테이션 계층 프레임워크 생성을 위한 전 단계로 프레임워크의 XML 문서화를 제안하였다. 이러한 문서의 장점은 웹 애플리케이션 프리젠테이션 프레임워크를 보기 쉽고 관리하기 쉽게 했다는 점 이외에도 본 XML 문서를 확장하여 현재 개발되어 있는 많은 종류의 웹 애플리케이션 서버 의존적인 부분까지도 프레임워크화 하여 소스코드를 자동 생성 해줄 수 있는 여지를 지니고 있다는 데 있다. 또한 개발자에게 보다 쉽게 웹 애플리케이션을 개발할 수 있게 하고 DBC 접근법이 적용된 XML 문서이기 때문에 DBC 접근법이 지니고 있는 신뢰성 있는 소프트웨어의 구축에 많은 도움을 줄 수 있을 것이다. 향후 연구로는 각 웹 애플리케이션 서버에 의존적인 부분을 중점적으로 연구하여 이러한 부분도 프레임워크화 해야 할 필요성이 있다. 또한 이 프레임워크를 이용하여 소스 코드를 자동 생성하기 위한 연구도

[1] Blue Print Team, "Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition," Addison Wesley, 2001.

[2] Blue Print Team, "Java Pet Store Demo Application," <http://java.sun.com/blueprints/>, 2001.

[3] P. Giotta, S. Grant, M. Kovacs, "Professional JMS Programming," Wrox, 2000.

[4] 노혜민, 유철중, 장옥배, "DBC 접근법을 이용한 엔터프라이즈 빈즈 명세 기법," 한국정보과학회 학술발표논문집(I), 제28권, 제2호, pp.421-423, 2001.

[5] ISE, "Building bug-free O-O software : An introduction to Design By Contract", <http://www.eiffel.com/doc/manuals/technology/contract/page.html>, 2000.

[6] 김운용, 최영근, "XML을 이용한 디자인 패턴 활용 구조," 한국정보처리학회 학술발표논문집(상), 제8권, 제2호, pp.335-338, 2001