

# 웹에서의 시나리오 편집기 구현

신기호, 박성은, 이용규  
동국대학교 컴퓨터공학과  
e-mail: kihos@dgu.ac.kr

## Implementation of a Scenario Editing Tool on the Web

Ki Ho Shin, Sung Eun Park, Yong Kyu Lee  
Dept. of Computer Engineering, Dongguk University

### 요약

기존의 문서 편집기는 시나리오 작성에 편리한 기능을 제공하지 않으므로 영화나 연극의 시나리오를 작성하는데 불편하였다. 또한 시나리오 작가가 웹 표준인 XML 형태로 시나리오를 작성하려 하여도 웹 기술에 익숙하지 않으므로 어려움이 많았다. 본 논문에서는 이러한 문제를 해결하기 위해서 시나리오 작가가 XML에 대해 모르더라도 웹을 통해 XML 기반의 시나리오를 쉽게 작성, 수정할 수 있으며, 쇼 기능을 통해 작성된 시나리오를 확인할 수 있는 웹 기반의 시나리오 편집 시스템을 구현한다. 이 시스템은 웹에서 XML 형태의 시나리오를 작성할 수 있어 표준화와 데이터 공유 등의 장점이 있다.

### 1. 서론

웹에서 XML 형태의 시나리오를 작성하는데 익숙하지 않은 작가는 XML 기반의 시나리오를 작성하는데 어려움이 많고, 이미 작성된 시나리오는 연습하는 도중에 수정하기가 어렵다.

본 논문에서는 이러한 단점들을 해결하기 위해 웹에서 XML 기반의 시나리오를 직접 작성하고 편집할 수 있는 시스템을 개발한다. 이 시스템은 차세대 웹 표준 언어인 XML(eXtensible Markup Language)을 이용하여 시나리오를 작성할 수 있으며, XSL을 이용하여 사용자에게 다양한 화면을 제공한다. 그리고, 기존의 데이터형의 제약이 많은 세익스피어 DTD[6]를 확장성이 뛰어난 XML 스키마로 변경하며, 시나리오 XML 문서를 파싱하기 위해 DOM, SAX를 이용한다.

본 논문에서는 XML 기술을 사용하여 사용자와 웹과 XML을 알지 못해도 XML 스키마 구조에 맞는 XML 기반의 시나리오를 작성할 수 있고, 작성된 시나리오를 시나리오 쇼 기능을 통해 확인할 수 있다. 또한, 시나리오 쇼 중간에 편집기능을 이용하여 시나리오를 수정할 수 있으며, 실시간으로 수정된 내용을 볼 수 있는 장점이 있다.

### 2. 관련 기술

웹에서의 시나리오 편집시스템을 구현하기 위해 XML 스키마, DOM, SAX, XSL, SOAP 기술을 사용한다.

#### 2.1 XML 스키마

XML 문서의 구조를 정의할 때 사용하기 편리한 DTD가 많이 사용되고 있다. 하지만 DTD는 지원하는 데이터 형식에 제한이 많고, 네임스페이스를 적용하기 어려웠다. 이와 같은 DTD의 제약점을 해결하기 위해 W3C에 의해 XML 스키마(XML Schema)[5]가 표준화되었다. XML 스키마는 다양한 데이터 형식을 지원하며, 구조가 XML 형태라는 장점이 있다.

<표 1> DTD와 XML 스키마 비교

	DTD	XML Schema
문서 형태	비 XML문서 형태	XML 문서 형태
데이터 형	기본적인 데이터형	다양한 데이터형
반복 지정	?, +, * 표현지원	최소, 최대 반복횟수 지원

#### 2.2 XSL

XML은 단지 정보를 표현하는데 사용되어지며

XSL(eXtensible Stylesheet Language)을 이용하면 다양한 형태로 XML 문서를 표현 할 수 있다. CSS와 다르게 XSL은 변환(Transformation)과 포매팅(Formatting)기능을 가지고 있다. 변환 기능을 사용하면 XML 문서에서 내용을 재구성하여 새로운 XML 문서를 생성할 수 있고, 포매팅 기능을 사용하면 XML 문서에 레이아웃이나 스타일 정보를 부여할 수 있다.

### 2.3 SOAP

SOAP(Simple Object Access Protocol)[4]은 기존의 분산된 객체들이 서로간에 통신하기 어려운 문제점을 해결하기 위해 W3C에서 표준화된 프로토콜이다. SOAP은 HTTP 80포트로 XML로 인코딩된 메시지를 주고받을 수 있고, 차세대 웹 서비스의 핵심 프로토콜이다.

### 3. 시나리오를 위한 XML 스키마

기존의 세익스피어 DTD[6]는 데이터형을 표현하는데 제약점이 많았다. 본 논문에서는 기존의 시나리오 DTD보다 확장성이 뛰어난 XML 스키마로 바꾸었다. 시나리오에는 <표 2>와 같은 요소들이 등장하며, 각 요소는 XML 기반의 시나리오를 위해 엘리먼트 명을 갖게 된다.

<표 2> 시나리오에 사용되는 엘리먼트

엘리먼트	설명
TITLE	시나리오의 제목
PREFACE	시나리오의 설명
ACTORS	등장 배우들
SCNDESCR	배경 설명
PLAYSUBT	시나리오의 줄거리
ACT	막, 장
SPEECH	대사 부분
SPEAKER	배우
LINE	대사
SUBLINE	지문

<표 2>에 나타난 시나리오 엘리먼트들은 [그림 1] 같은 형태의 XML 스키마에 의해 정의된다. 그리고, 최상위 엘리먼트는 <PLAY>이며 서브 엘리먼트에는 <표 2>에서 나열한 엘리먼트들이 차례로 나타난다. 이러한 전체적인 구조에서 각 엘리먼트들은 "ref"라는 속성을 통해 XML 스키마의 다른 부분에서 정의된다.

DTD와는 달리 XML 스키마는 엘리먼트의 중복 횟수를 설정하고 있다. 시나리오에서 여러 번 등장하는 막·장 태그인 "ACT"는 [그림 1]에서와 같이 minOccurs와 maxOccurs 속성을 통해 반복 횟수를

지정할 수 있다.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PLAY">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="TITLE" />
        <xs:element ref="PREFACE" />
        <xs:element ref="ACTORS" />
        <xs:element ref="SCNDESCR" />
        <xs:element maxOccurs="unbounded" ref="ACT" />
        ...
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  ...
</xs:schema>
```

[그림 1] 시나리오 XML 스키마 전체 구조

막·장 태그인 "ACT"는 "ref" 속성을 이용하여 XML 스키마에서 정의된 구조가 [그림 2]에 나타나 있다. "ACT" 태그는 "SUBTITLE," "STAGEDIR," "SCENE," "SPEECH"로 구성되어 있으며 각 엘리먼트 "ref" 속성을 이용하여 XML 스키마의 다른 부분에서 정의된다.

```
<xs:element name="ACT">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SUBTITLE" />
      <xs:element minOccurs="0" ref="STAGEDIR" />
      <xs:element maxOccurs="unbounded" ref="SCENE" />
      <xs:element maxOccurs="unbounded" ref="SPEECH" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

[그림 2] 엘리먼트의 XML 스키마 예

[그림 1]과 [그림 2]에서 정의한 XML 스키마 구조에 맞는 XML 기반의 시나리오는 [그림 3]과 같이 나타난다. 예를 들어, <ACT> 태그는 XML 스키마 구조에서 정의된 것처럼 횟수에 관계없이 반복될 수 있다. 또한 다른 태그들도 XML 스키마 구조에 맞게 시나리오에서 사용된다.

```
<?xml version="1.0" encoding="euc-kr"?>
<PLAY>
  <PREFACE>
    <P>人形劇(인형극) 나무꾼과 선녀 </P>
  </PREFACE>
  ...
  <ACT>
    <SUBTITLE>1막 1장 </SUBTITLE>
    <STAGEDIR>나무꾼의 집</STAGEDIR>
    <SCENE>무대 왼쪽에 커다란 크리스마스주리. 쓸쓸하다. </SCENE>
    <SPEECH>
      <SPEAKER>기호</SPEAKER>
      <SUBLINE>(객석을 향하여) </SUBLINE>
      <LINE>이제 겨우 잠들이 들었구나</LINE>
    </SPEECH>
  </ACT>
  ...
</PLAY>
```

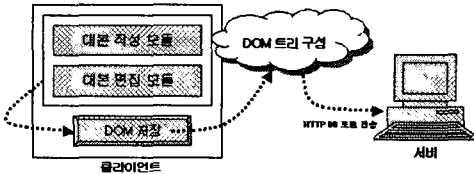
[그림 3] XML 기반의 시나리오

4. 시나리오 편집 시스템 개요

시나리오 편집 시스템에서는 작가가 시나리오를 작성하거나 수정할 수 있다. 작가는 XML 스키마의 구조를 알지 못하더라도 XML 기반의 시나리오를 작성할 수 있다.

4.1 DOM을 이용한 시나리오 편집

시나리오를 작성하고 편집하기 위해서는 XML 스키마 구조에 따라야 한다. 시나리오 편집 모듈에서는 작가가 입력한 데이터를 XML 스키마 구조에 맞는 XML 기반의 시나리오 문서를 생성하고, DOM 트리를 구성한다. [그림 4]와 같이 시나리오 편집 시스템은 클라이언트에서 DOM을 이용하여 시나리오를 작성하는 것을 보여주고 있다.

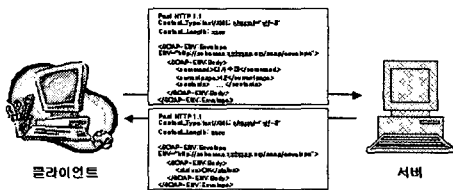


[그림 4] DOM을 이용한 시나리오 편집

작가가 입력한 데이터는 DOM 트리 구조로 구성되고, SOAP 메시지로 인코딩되어 HTTP 80포트의 POST 방식으로 서버에 전달한다.

4.2 SOAP을 이용한 데이터 전송

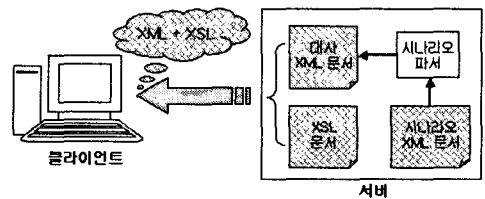
SOAP은 HTTP의 80포트를 이용하기 때문에 방화벽을 쉽게 통과 할 수 있고, XML 형태의 메시지를 가지고 있으므로 플랫폼에 상관없이 메시지를 이해할 수 있다. [그림 5]는 클라이언트와 서버간에 SOAP 메시지를 주고받는 것을 나타내고 있다. 우선 클라이언트는 서버에 대사의 수정을 요구하는 SOAP 메시지를 보내고, 서버에서는 수정된 결과를 클라이언트에 SOAP 메시지로 되돌린다.



[그림 5] SOAP 메시지 전송

4.3 XSL을 이용한 시나리오 쇼

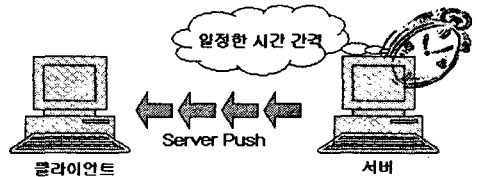
시나리오 작성 시스템을 이용하여 XML 기반의 시나리오를 만든 후, 시나리오 쇼 모듈을 이용하여 작성된 시나리오를 확인할 수 있다. [그림 6]과 같이 서버에는 XML 기반의 시나리오와 XSL이 저장되어 있다. 클라이언트가 시나리오 쇼를 요청하면 파서는 시나리오에서 순차적으로 대사를 추출하여 XML 대사 문서를 생성한다. 대사 XML 문서는 XSL과 함께 클라이언트에 보내지고 클라이언트는 대사를 화면에 보여준다.



[그림 6] XSL을 이용한 시나리오 쇼 모듈

4.4 푸시(Push)를 이용한 쇼

클라이언트는 [그림 6]과 같이 XML과 XSL을 화면에 보여준다. 이러한 대사 XML과 XSL을 화면에 쇼 형태로 보여주기 위해서는 서버 푸시가 필요하다. [그림 7]은 서버에서는 대사를 일정한 시간 간격마다 클라이언트로 보내는 서버 푸시를 나타낸 것이다



[그림 7] 서버 푸시를 이용한 쇼

시나리오 쇼는 "PLAY," "STOP," "REPLAY"와 같은 기능이 필요하다. 사용자는 시나리오 쇼를 진행하는 중간에 멈추거나, 원하는 대사의 위치로 이동할 수 있다.

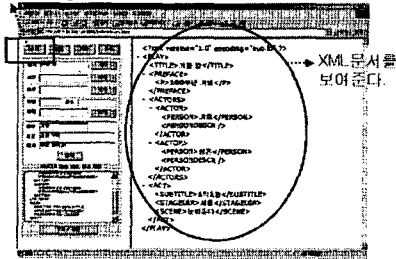
5. 시나리오 편집 시스템 구현

본 논문에서 구현한 시나리오 편집 시스템은 시나리오를 작성하고 편집할 수 있는 부분과 작성된 시나리오를 쇼 형태로 클라이언트에게 보여주는 기능을 가지고 있다.

### 5.1 시나리오 작성 및 편집 모듈 구현

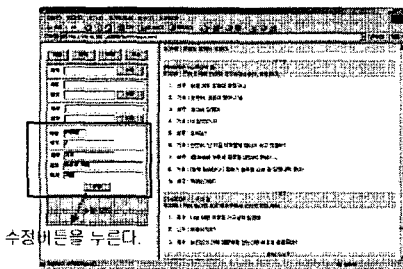
시나리오 작성 모듈을 구현하기 위해 클라이언트에는 애플릿, 자바스크립트와 MSXML을 이용한다. [그림 8]에서 작가가 각 입력창에 원하는 데이터를 입력하면 애플릿 내부에서 XML 스키마 구조에 맞는 XML 기반의 시나리오를 작성하게 된다. 그리고, MSXML의 DOM 객체와 자바스크립트를 사용하여 오른쪽 프레임에 작성중인 시나리오를 표시한다.

수정단계로 전환



[그림 8] 시나리오 작성 시스템 구현

시나리오 쇼 중간에 시나리오를 수정하고 싶을 경우에 사용자는 편집 모듈로 전환할 수 있다. [그림 9]는 이러한 편집 모듈을 보여주고 있다. 자바스크립트를 이용하여 XML 문서에 XSL을 적용하여 보여주면 사용자는 애플릿으로 구현된 편집기능을 이용하여 시나리오를 수정할 수 있다.



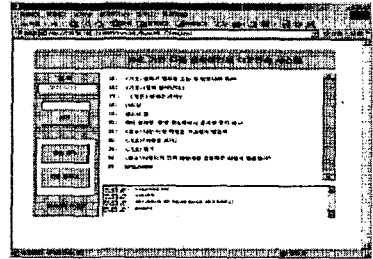
[그림 9] 시나리오 수정 모듈 구현

수정하고자 하는 부분을 입력하고 수정버튼을 누르면 데이터가 서버의 JSP로 전송된다. 그리고, 시나리오를 수정하고 응답 메시지를 애플릿에 보낸다. 애플릿은 자바스크립트 함수를 호출하여 서버에 있는 수정된 시나리오를 다시 로드하게 한다.

### 5.2 시나리오 쇼 모듈 구현

사용자는 작성된 시나리오를 쇼 기능을 통해 작성된 시나리오를 확인해 볼 수 있다. [그림 10]에서 처

럼 사용자는 애플릿으로 구현한 시나리오 쇼 기능을 이용해 시나리오를 확인해 볼 수 있다. 사용자는 쇼를 보는 중간에 시나리오를 수정하기 원할 경우 애플릿의 스테드를 이용하여 [그림 9]와 같은 수정모듈로 전환할 수 있다.



[그림 10] 시나리오 쇼 모듈 구현

## 6. 결론

본 논문에서는 웹에서 XML 기반의 시나리오를 작성할 수 있고, 쇼 기능을 이용하여 시나리오를 확인할 수 있는 시나리오 편집기를 개발하였다.

이 시스템은 시나리오의 데이터만 표현되는 XML 문서에 다양한 XSL을 적용하여 여러 스타일의 화면을 볼 수 있게 하였고, 작가가 시나리오 쇼 기능을 사용하여 작성된 시나리오를 확인할 수 있으며, 잘못된 내용은 편집 모듈을 이용하여 실시간으로 수정할 수 있다.

따라서, 웹과 XML에 익숙하지 않은 시나리오 작가도 본 논문에서 구현한 시나리오 편집기를 이용하여 웹에서 XML 형태의 대본을 쉽게 작성할 수 있고, 표준화와 데이터 공유 등의 장점이 있다.

## 참고 문헌

- [1] Manfred Hauswirth and Mehdi Jazayeri, "A Component and Communication Model for Push Systems," ACM, September 6-10, 1999.
- [2] Brett McLaughlin, "Java & XML," 2nd Ed. O'REILLY, 2000.
- [3] Alexander Nakhimovsky and Tom Myers, "Professional Java XML Programming," XROX Press, 2000.
- [4] SOAP 1.2 Spec., <http://www.w3.org/TR/SOAP>
- [5] XML 스키마 Spec., <http://www.w3.org/XML/Schema>
- [6] 셰익스피어 DTD, <http://www.oasis-open.org/cover/bosakShakespeare200.html>