

XML 오브젝트의 가시화

김혜연^U 조동섭
이화여자대학교 컴퓨터학과
{992COG09, dscho}@mm.ewha.ac.kr

Visualization of XML Object

Hye-Yeon Kim^U Dong-Sub Cho
Dept. of Computer Science, EWHA Womans University

요 약

XML data를 VRML을 사용하여 시각적으로 나타내는 방법을 연구하였다. 현재 Web 환경은 동적으로 문서를 생성하고 사용자가 보기 쉽게 그래픽으로 표현하는 방향으로 발전하고 있다. 이러한 환경에서 XML은 실시간으로 data를 생성하기 쉬워 많이 사용되고 있으나 text 기반이기 때문에 data를 가시화하여 사용자한테 보여주기 힘들다는 단점이 있다. 이에 VRML을 XML과 결합하여 실시간으로 변하는 data를 VRML과 같은 시각화 도구를 사용하여 표현하는 방법에 대해 연구를 하였다. 본 논문에서는 Java Servlet을 사용하여 XML 문서에서 data를 추출하여 VRML 코드를 만들고, 그 코드를 사용자측에 전달하여 시각적으로 data를 볼 수 있도록 하였다.

1. 서론

현재 Web 환경은 동적으로 문서를 생성하고 사용자가 보기 쉽게 그래픽으로 표현하는 방향으로 발전하고 있다[3]. 이러한 환경에서 XML은 실시간으로 data를 생성하기 쉬워 많이 사용되고 있으나 text 기반이기 때문에 data를 가시화하여 사용자한테 보여주기 힘들다는 단점이 있다. 이에 VRML을 XML과 결합하여 실시간으로 변하는 data를 VRML과 같은 시각화 도구를 사용하여 표현하는 방법이 유용하다. 본 논문에서는 Java Servlet을 사용하여 XML 문서에서 data를 추출하여 VRML 코드를 만들고, 그 코드를 사용자측에 전달하여 시각적으로 data를 볼 수 있도록 하였다.

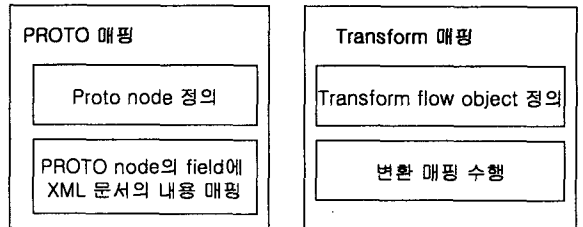
본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 기존 연구들을 살펴보고 3장에서는 본 논문이 제안하고자 하는 시스템에 대해 설명하고 4장에서는 결론 및 향후 방향에 대해 살펴본다.

2. 관련연구

2.1 Integrating XML and VRML: A Technical Discussion[1]

XML의 data를 VRML로 바꾸기 위해 XSL 스타일 시트를 사용한다. PROTO node를 정의하고 XML 파일

의 내용이 PROTO node의 field로 어떻게 매핑되는지 정의한다.[6]이 방식은 XML의 data 중에서 "ID" 속성을 가진 요소들에 접근하여 이 data를 시각적으로 표현하도록 하였다. 즉 ID 속성을 가진 XML data에만 접근할 수 있다는 제한을 부과하여 구현되었다. XML로 VRML 코드를 만들기 위해 다음과 같은 두 가지 방식을 사용한다.[1]



2.1.1 PROTO Mappings

XSL파일에 PROTO라 불리는 flow 오브젝트를 정의하여 PROTO node가 결과 코드로 만들어지게 한다. 그리고 XML 파일의 내용이 PROTO 노드의 필드에 어떻게 매핑될 것인지 정의한다. 즉 </children>을 써서 XML 문서의 text의 내용을 가져와 그것을 PROTO노드의 필드에 매핑시킨다.

XML 코드로 PROTO Mapping을 하는 과정은 다음과 같다.

```
<?xml version="1.0"?>
<stocks>
<stock id="ORCL">
<symbol>ORCL</symbol>
<growth>10</growth>
<performance>9</performance>
<pe>5</pe>
</stock>
</stocks>
```

다음과 같은 PROTO node를 정의한다.

```
PROTO bar [
  field SFInt32 x 0
  field SFInt32 y 0
  field SFInt32 z 0
  field MFString label [ "" ]
]
```

다음 XSL은 PROTO node의 field에 </children>을 사용하여 XML의 문서의 내용을 매핑한다.

```
<rule>
  <target-element type="stock"/>
  <PROTO>bar</PROTO>
  <children/>
</rule>

<rule>
  <element type="stock">
    <target-element type="growth"/>
  </element>
  x <children/>
</rule>
```

결과로 다음 VRML 노드가 만들어진다.

```
bar {
  x 10
  y 9
  z 5
  label "ORCL"
}
```

2.1.2 Transformation Mappings

생성될 VRML 문서에 Transform 노드들을 만든다. 모든 변환 정보들은 사용자의 PROTO에 명시적으로

정의되어야 한다. 새로운 TRANSFORM flow object를 정의하여 변환 정보들이 PROTO node에 나타내도록 해줘야 한다.

XML을 3차원 icon을 사용해서 그래픽하게 나타내는 과정은 다음과 같다.

```
<?xml version="1.0"?>
<people>
<person>
  <name>Smith</name>
  <salary>1500</salary>
  <years>14</years>
</person>
<person>
  <name>Jones</name>
  <salary>1800</salary>
  <years>12</years>
</person>
</people>
```

각 icon은 x 축을 따라 상수 값을 가져야 한다. 이것을 위해 다음 XSL을 사용하여 값을 변환하여 할당해준다. 즉 다음 코드를 수행하면 각 pson의 translation의 x값이 increment_value인 10씩 증가하여 할당된다.

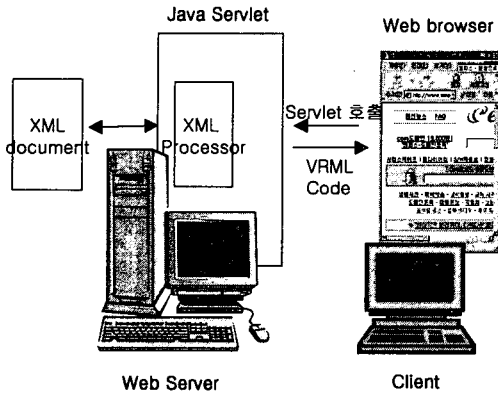
```
<rule>
  <target-element type="person"/>
  < T R A N S F O R M
    increment_field="translation.x"
    increment_value=10>
  <PROTO>Emplcon</PROTO>
  <children/>
</TRANSFORM>
</rule>
```

결과로 다음 VRML 코드가 만들어진다. translation의 x가 10씩 증가한 것을 알 수 있다.

```
Transform {
  translation 0 0 0
  children [
    Emplcon {
      name [ "Smith" ]
      salary [ "$1500" ]
      years 10 14 10
    }
  ]
}
Transform {
  translation 10 0 0
  children [
    .....
  ]
}
```

3. XML 오브젝트의 가시화

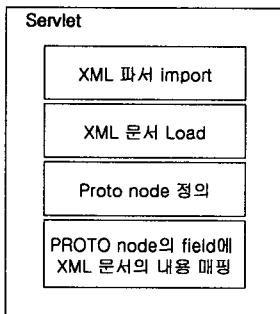
3.1 시스템 구성



[그림 1] 시스템 구성도

그림 1은 본 논문에서 제안하는 시스템의 구성도이다. Client의 요청에 의해 서버측에 있는 Java Servlet이 호출되면 Servlet에 포함된 XML Processor가 XML 문서를 가져와서 Parsing한 후 필요한 요소를 가져와 가시화하기 위한 VRML code로 만들어주는데 사용한다. 만들어진 VRML 코드는 Client에 전달되어 그래픽적으로 표현된다.

3.2 Servlet 프로그램



[그림 2] Servlet 프로그램의 흐름도

그림 2는 본 시스템의 핵심 부분인 Servlet 프로그램의 흐름도이다[4][5]. Servlet 프로그램은 XML문서를 다루기 위해 XML 파서의 class를 import한다. 그리고 시각화하고자하는 XML 문서의 위치를 명시하여 XML 문서를 가져온다. 그리고 VRML 변환을 하기 위한 Proto node를 정의한 후 XML parser의 Class에서 제공해주는 함수를 하여 Proto node에 XML 문서의 내용을 매핑한다. 결과는 VRML로 바뀌어 Client측의 Web Browser

로 전송된다.

4. 결론 및 향후 연구

XML data를 VRML을 사용하여 시각적으로 나타내는 방법을 연구하였다. VRML을 XML과 결합하여 실시간으로 변하는 data를 VRML과 같은 시각화 도구를 사용하여 표현하였다. 즉 Java Servlet을 사용하여 XML 문서에서 data를 추출하여 VRML 코드를 만들어 사용자측에서 시각적으로 data를 볼 수 있도록 하였다. 2장에서 살펴본 XSL을 사용하여 VRML을 생성하는 방식은 XML문서의 ID 속성을 가진 요소들에만 접근하여 코드를 생성할 수 있다는 제약점이 있다. 그런데 본 시스템은 Java Servlet으로 구현하여 XML 문서의 모든 요소에 접근 가능하며 서버측에서 코드를 생성해서 결과만 클라이언트한테 전달해 주므로 네트워크 오버헤드도 줄일 수 있다. 또한 XML의 모든 data를 클라이언트 측에 전달하지 않고 한정된 data만 전달하기 때문에 정보 보호 측면에서도 좋다.

앞으로는 XML data를 쉽게 VRML로 바꾸는 저작물을 만들고, XML 문서를 Database에 연결하여 더 다양한 문서를 시각화할 수 있도록 시스템을 확장할 것이다.

참고 문헌

- [1] <http://www.vrml.org/WorkingGroups/dbwork/vrmlxml1.html>, 1998
- [2] Frand Boumphrey 외 11인, "Professional XML Applications", WROX, 1999
- [3] Neil Bradley, "The XML Companion", Addison Wesley, 2000
- [4] Hiroshi Maruyama 외 2인, "XML and Java Developing Web Applications", Addison Wesley, 1999
- [5] Michael Leventhal 외 2인, "Designing XML Internet Applications", Prentice Hall PTR, 1998
- [6] Kris Jamsa, Ph.D. Phil Schmauder Nelson Yee 저 김숙자 옮김, VRML 프로그래머 라이브러리, 성안당, 1997