2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

# XML 기반 차트 편집 및 디스플레이 시스템 구현

윤 현 님 · 이 용 규
동국대학교 컴퓨터공학과

# Implementation of a Chart Editing and Display System Based on XML

Hyun Nim Yoon · Yong Kyu Lee
Dept. of Computer Engineering, Dongguk University

**요 약**

웹의 사용자가 늘어나면서 웹을 통해 정보를 표현하고 제공하는 수단으로 차트의 사용이 많아지고 있다. 그러나 기존의 차트 표현 방법들은 표현 방법과 구입비용 그리고 호환에 대한 문제점들을 갖고 있다. 따라서 본 논문에서는 이를 해결하기 위해 차트 정보를 웹 표준 언어인 XML을 이용하여 쉽게 표현하고 디스플레이 하는 방법을 제시한다. 본 논문에서 정의한 차트 작성 언어인 CML은 기존의 XML 기반 그래픽 표현 애플리케이션의 복잡성과는 달리 막대형, 원형, 선형의 이차원 차트 정보를 간결하게 표현하고 브라우저를 이용하여 디스플레이 할 수 있다. 본 논문에서 정의한 XML 차트 표현은 어느 플랫폼에서도 호환 가능하며 웹 표준에 근거하였으므로 웹에서의 차트 정보의 공유가 가능하다.

## 1. 서론

인터넷 상에서 정보의 시각적 표현 수단으로 차트의 사용이 많아지고 있다. 그러므로 웹 상에서 차트를 사용하여 정보를 표현하는데 어려움을 최소화해야 한다. 현재 차트의 작성에 많이 사용되고 있는 대표적인 프로그램에는 마이크로소프트사의 엑셀[1]과 MathWorks사의 MATLAB[2]을 들 수 있다. 엑셀은 3차원 구조의 워크시트로 윈도우의 특징인 WYSIWYG의 형태를 갖고 있으며 편리한 수식 계산 및 다양한 기능을 발휘한다. MATLAB(MATrix LABoratory)은 MathWorks사에서 개발한 언어로서 다양한 활용 및 그래픽 및 차트 표현 기능 등을 제공한다. 그러나 이들은 차트 표현의 상호 호환성, 웹 상에서 구현의 문제점을 갖고 있다.

현재 웹 상에서 차트를 작성하거나 디스플레이 할 수 있는 적절한 시스템은 알려져 있지 않으나, 그래픽 처리를 위하여 개발된 XML 기반의 애플리케이션들을 사용할 수 있다. VML(Vector Markup Language)[6]은 벡터 방식을 이용하여 웹 그래픽의 대부분을 이루는 일반적인 비트맵보다 훨씬 더 유연하고 편집이 용이하다. 그러나 그래픽 요소를 표현하는데 필요한 속성의 사용이 복잡하고 어려운 단점이 있다. PGML(Precision Graphics Markup Language)[4]은 기존의 플랫폼들에서 웹으로의 확장을 위해 애니메이션과 다이나믹한 처리를 위한 객체모델을 포함하지만 이진 파일 형식을 채택하여 편집하기가 불가능하다. SVG(Scalable Vector Graphic)[5]는 비트맵에 기반을 둔 웹 이용 그래픽 표현을 위해 많은 장점들을 제공한다. 그래픽 표기법에서 그래픽 정보를 부호화하는 반면 그래픽 개체의 형식을 지원하는 뷰어가 필요하므로 범용 웹 브라우저에서 디스플레이가 되지 않는다. 이러한 그래픽 처리를 위한 애플리케이션들은 XML을 기반으로 하였으므로 웹 상에서 차트 구현의 가능성은 있지만 다양한 그래픽에 대한 많은 표현 요소와 속성을 갖고 있기 때문에 그래픽 처리의 적용범위는 넓은 반면 차트의 구현을 위한 사용에는 복잡하고 필요한 요소를 분류해내기가 어렵다. 그러므로 차트의 표현을 위한 필수적인 요소와 간단한 속성만으로 구성된 XML 기반 차트 표현 애플리케이션이 필요하다.

따라서 본 논문에서는 기존 차트 작성 프로그램과 그래픽 처리를 위한 XML 기반 애플리케이션의 장단점을 고려하여 웹 표준언어인 XML을 사용하면서 쉽게 차트 정보를 표현하고 디스플레이할 수 있는 차트 작성 시스템을 개발한다. 막대형, 원형, 선형의 이차원 차트를 작성할 수 있으며 디스플레이를 위한 별도의 프로그램이 필요없이 기존의 웹 브라우저만을 사용하여 차트의 출력이 가능하도록 한다.

## 2. CML(Chart Markup Language)

XML을 이용한 차트의 표현을 위해서 CML(Chart Mark up Language)을 정의하고 차트의 표현 정보를 출력하기 위

한 방법을 제시한다. 기존의 XML 기반 그래픽 처리 애플리케이션의 복잡성을 탈피하여 차트의 구현에 필수적인 요소만으로 정의하였으므로 차트의 구현이 쉽고 간단하다. 이 절에서는 차트 구현을 위한 CML DTD와 이에 기반하여 작성한 CML 문서를 제시하고 간편한 CML 문서 작성을 위한 CML 문서 편집기를 소개한다.

### 2.1 차트의 기본 요소

CML DTD에서 표현하여야 하는 주요 차트 정보를 정리하면 <표 1>과 같다.

<표 1> CML DTD에서 표현하는 주요 차트 정보

| 엘리먼트 | 에트리뷰트 | 기능 설명 |
|---|---|---|
| chart | 없음 | 차트의 표현 |
| charttype | shape,planetype | 차트 종류와 기본평면 유형 |
| chartinfo | 없음 | 차트 표현을 위한 요소 |
| header | headerposition | 차트의 제목 |
| yunit | 없음 | 데이터 단위 표현 |
| title | 없음 | 막대 차트의 축 제목 정보 |
| xtitle | position | 막대 차트의 항목 축 제목 |
| ytitle | position | 막대 차트의 값 축 제목 |
| axisinfo | 없음 | 막대 차트의 축 표현 정보 |
| ymax | 없음 | 막대 차트의 값 축 표현을 위한 최대값 |
| ymin | 없음 | 막대 차트의 값 축 표현을 위한 최소값 |
| xaxis | 없음 | 막대 차트의 항목 축 표현 정보 및 항목 |
| item1 item2 item3 item4 item5 | color | 항목 |
| legend | 없음 | 범례 정보 |
| legend1 legend2 legend3 legend4 legend5 | color | 각 항목에 대한 범례 |
| value | 없음 | 데이터 정보 |
| value1 value2 value3 value4 value5 | color | 각 항목의 데이터 |

### 2.2 CML DTD의 정의

CML은 차트의 정보를 마크업하기 위한 다양한 논리적 정보를 포함하고 있으며 기존 HTML 사용자들도 쉽게 사용할 수 있을 정도의 간결한 구조를 제공하고 있다. (그림 1)은 CML DTD의 일부이다.

```
<!ENTITY % att-preinfo    'shape (circle|hbar|vbar) #IMPLIED'>
<!ENTITY % att-plane      'planetype (1|2)      #IMPLIED' >
<!ENTITY % att-position   'position (center)     #IMPLIED'>

<!ELEMENT   CML   (chart) >

<!ELEMENT   chart   ((charttype*), chartinfo?) >

<!ELEMENT charttype   EMPTY >
<!ATTLIST charttype %att-preinfo; %att-plane;>

<!ELEMENT chartinfo (header,yunit?,title?,axisinfo?,value,legend) >
-------------------------------이하 생략-------------------------------
```

(그림 1) CML DTD의 일부

### 2.3 CML 문서의 작성

차트의 정보를 표현하기 위해서는 CML 문서 인스턴스를 작성해야 한다. <표 2>는 본 논문에서 CML 문서 인트턴스를 작성하는데 사용될 데이터를 표로 작성한 것이다. 이 데이터는 CML 각 요소의 내용부분에 해당되며 (그림 2)와 같은 CML 문서 인스턴스를 작성하게 된다. CML 문서 인스턴스는 사용자가 직접 작성하는 것이 아니라 다음 절에서 설명할 CML 문서 편집기를 이용하여 자동으로 작성하게 된다.

<표 2> 차트 데이터 테이블

| | Korean | English | Mathematics | Society | Science |
|---|---|---|---|---|---|
| Class 1 | 90 | 70 | 70 | 85 | 55 |
| Class 2 | 95 | 90 | 69 | 75 | 65 |
| Class 3 | 100 | 80 | 80 | 95 | 60 |
| Total | 285 | 240 | 219 | 255 | 180 |
| Average | 95 | 80 | 73 | 85 | 60 |

다음 (그림 2)는 <표 2>의 각 과목에 대한 반별 평균을 쉽게 비교하기 위한 목적으로 막대 차트로 표현하기 위하여 작성한 CML 문서 인스턴스이다.

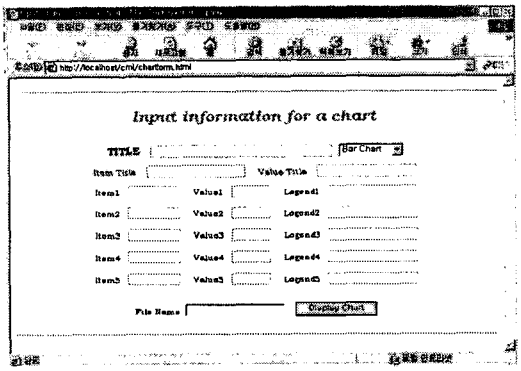
```
<CML>
 <chart>
   <charttype shape="vbar" planetype="1"/>
      <chartinfo>
         <header headerposition="up">A Midterm Examination</header>
         <title>
            <xtitle position="center">Subject</xtitle>
            <ytitle position="center">Average</ytitle>
         </title>
         <axisinfo>
            <ymax>100</ymax>
            <ymin>0</ymin>
            <xaxis>
               <item1>Kor</item1>
               <item2>Eng</item2>
               <item3>Mat</item3>
               <item4>Sci</item4>
               <item5>Soc</item5>
            </xaxis>
         </axisinfo>
         <value>
            <value1 color="#FFFFCC">95</value1>
            <value2 color="#FFCCFF">80</value2>
            <value3 color="#CCFFFF">73</value3>
            <value4 color="#CCCCFF">85</value4>
            <value5 color="#CCCCCC">60</value5>
         </value>
         <legend>
            <legend1 color="#FFFFCC">Korean</legend1>
            <legend2 color="#FFCCFF">English</legend2>
            <legend3 color="#CCFFFF">Mathematics</legend3>
            <legend4 color="#CCCCFF">Science</legend4>
            <legend5 color="#CCCCCC">Society</legend5>
         </legend>
      </chartinfo>
 </chart>
</CML>
```

① (value 블록 전체)

(그림 2) CML 문서의 예

### 2.4 CML 문서 편집기

간단하고 쉽게 CML 문서를 작성할 수 있도록 CML 문서 편집기를 통한 인터페이스를 제공한다. 이것은 HTML의 입력양식을 이용해 차트 정보를 표현한다. 사용자가 입력한 정보는 ASP를 통하여 (그림 2)와 같은 CML 문서로 저장되며 저장 결과는 웹 브라우저를 통해 확인한다. 다음 (그림 3)은 CML 문서 편집기의 화면이다.




(그림 3) CML 문서 편집기 화면

### 3. CML 문서의 디스플레이

편집기를 통하여 작성된 차트 표현 정보는 브라우저를 통해 출력할 수 있어야 한다. 본 논문에서는 CML 문서를 출력하기 위하여 웹 브라우저를 사용하였다. 또한 CML DTD와 문서 인스턴스를 사용하여 차트를 쉽게 웹 브라우저 상에 디스플레이할 수 있다. 차트 디스플레이 시스템은 Windows 98 운영체제에서 구현하였으며 그래프의 출력을 위하여 마이크로소프트사의 XML 파서, ECMA DOM 스크립트, 인터넷 익스플로러 5.0을 이용하였다.

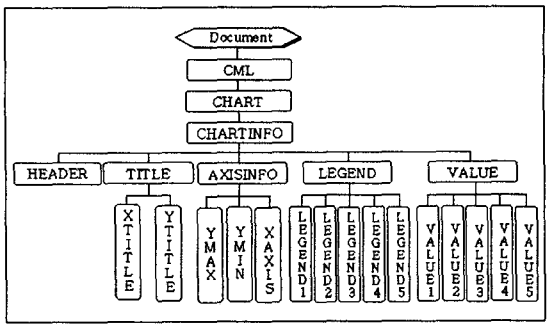
CML 문서는 XML 기반으로 문서의 구조적인 정보만을 저장하기 때문에 브라우저에 직접 출력할 수 없다. 그러므로 DOM을 이용하여 CML 문서를 VML 문서로 변환하고 DHTML의 레이어를 이용하여 브라우저에 출력한다. VML은 기존 비트맵 방식과 달리 곡선과 직선을 수학적 알고리즘으로 표현하므로 그래픽 구성이 훨씬 간단해 파일 크기와 출력속도를 기존 비트맵 방식보다 줄일 수 있다.

#### 3.1 DOM을 이용한 CML 문서의 VML 문서로의 변환

CML 문서의 차트 표현 정보를 디스플레이하기 위해서는 DOM을 이용하여 VML 문서로 변환하는 과정이 필요하다.

DOM(Document Object Model)[7]은 HTML과 XML 문서를 다루기 위한 W3C의 트리 기반 API 표준으로 플랫폼 및 언어 중립적인 인터페이스를 제정한 것이다. DOM을 이용하면 XML 문서를 객체화하여 이를 DOM 트리 형태로 접근할 수 있다. CML 문서의 차트 표현 정보를 디스플레이하기 위해서는 DOM을 이용하여 변환하는 과정이 필요하다. DOM 인터페이스를 통하여 CML 문서로부터 DOM 트리를 생성하고 CML 노드를 순회하며 HTML 문서내의 해당 VML 코드로 변환하게 된다.

(그림 2)의 CML 문서를 DOM 스크립트를 통해 DOM 트리로 변환한 결과는 다음 (그림 4)와 같다.




(그림 4) (그림 2)의 CML 문서를 DOM 트리로 변환한 결과

다음 (그림 5)는 막대 차트의 각 항목 데이터를 표현한 문서의 일부인 (그림 2)의 ①을 VML로 변환하기 위한 DOM 스크립트의 일부이다.

```
else if(x(i).nodeType==1 && x(i).nodeName=='value1'){
  if ((x(i).getAttribute('color'))=="#FFFFCC") {
    L=L+195
    T=T+187
    A1=((x(i).firstChild.nodeValue/10)*20);
    document.write("<v:rect fillcolor='#FFFFCC'style='position:absolute; top:"
    + T + "px;left:" + L + "px;width:40px;height:" + A1 +"px'>")
    document.write("</v:rect>")
    getchildren(x(i));}
    T-=187
  }
else if(x(i).nodeType==1 && x(i).nodeName=='value2') {
    if ((x(i).getAttribute('color'))=="#FFCCFF") {
    L+=67
    T+=217
    A2=((x(i).firstChild.nodeValue/10)*20);
    document.write("<v:rect  fillcolor='#FFCCFF'style='position:absolute;top:"
    + T + "px;left:" + L + "px;width:40px;height:" +  A2 +"px'>")
    document.write("</v:rect>")
    getchildren(x(i));
    }
    T-=217
}
----------------------------------이하 생략----------------------------------
```

(그림 5) CML 문서를 VML로 변환하기 위한 DOM 스크립트의 일부

다음 (그림 6)은 앞에서 설명한 과정을 거쳐 변환된 VML 문서의 일부이다.

```
<v:rect style="position:absolute;top:195px;left:187px;width:40px;height:95px"
fillcolor="#FFFFCC">
</v:rect>
<v:rect style="position:absolute;top:262px;left:217px;width:40px;height:80px"
fillcolor="#FFCCFF">
</v:rect>
----------------------------------이하 생략----------------------------------
```

(그림 6) VML 문서의 일부

#### 3.2 레이어를 이용한 CML 문서의 출력

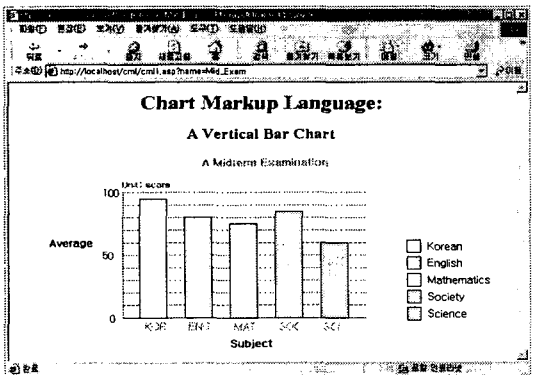
앞에서 변환된 VML 문서는 브라우저를 이용하여 출력이 가능하다. 그러나 본 논문에서는 출력 문서의 모양을 보다 좋게 하기 위하여 추가적으로 DHTML의 레이어를 이용하여 엘리먼트마다 자신의 위치에 출력한다. 레이어를 이용

한 차트의 출력은 드로잉 방식보다 빠른 출력시간을 얻을 수 있다. (그림 7)은 차트의 기본 영역을 표현하기 위해 레이어를 사용한 예의 일부이다.

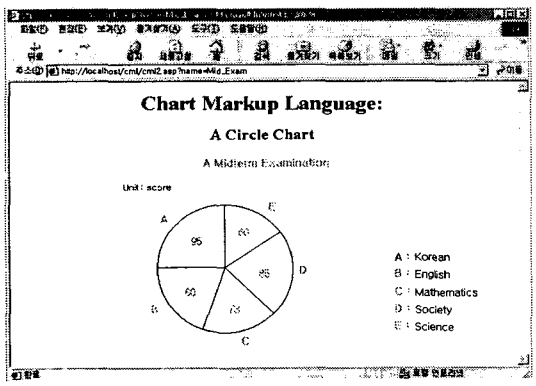
```
chart_basis_first_position += 15
document.write("<v:line style='position:absolute;' from=120pt," +
chart_basis_first_position + "pt to=390pt," +  chart_basis_first_position
+ "pt strokecolor=red strokeweight=0.5pt> ")
document.write("<v:stroke dashstyle=solid />")
document.write("</v:line>")
K1=chart_basis_first_position
K2=chart_basis_first_position+150
document.write("<v:line style='position:absolute:" + " from=120pt," + K1
+ "pt to=120pt," + K2 + "pt  strokecolor=red strokeweight=0.5pt> ")
----------이하 생략----------
```

(그림 7) 레이어를 사용한 예의 일부

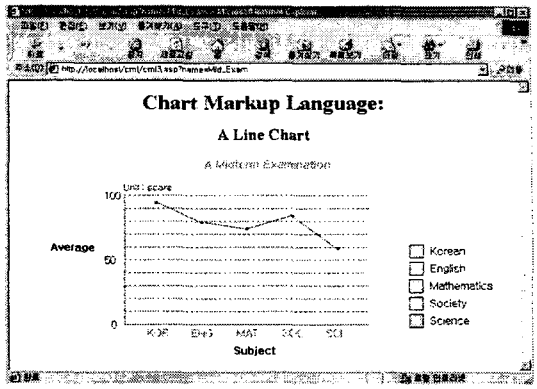
작성된 CML 문서를 디스플레이하기 위하여 웹 브라우저를 사용하였다. 웹 브라우저를 사용함으로써 별도의 출력 애플리케이션 없이도 사용자가 간단하게 차트를 디스플레이 할 수 있으며 인터넷 익스플로러 5.0에 DOM API를 지원하는 프로세서가 내장되어 있기 때문에 CML 문서를 쉽게 변환할 수 있다. 다음은 <표 2>의 차트 데이터를 이용하여 작성된 막대형, 원형, 선형의 이차원 차트의 CML 문서를 웹 브라우저에 각각 출력한 결과이다.




(그림 8) 브라우저에 막대형 차트 출력




(그림 9) 브라우저에 원형 차트 출력




(그림 10) 브라우저에 선형 차트 출력

## 4. 결론 및 향후 연구

차트의 표현 및 디스플레이를 위한 기존의 차트 정보를 표현하는 방법들의 문제점들을 해결하기 위해 XML 기반의 CML(Chart Markup Language)을 정의하였다. 범용 웹 브라우저를 통해 차트의 정보를 전달받을 수 있으므로 별도의 프로그램이나 호환기 없이도 디스플레이가 가능하다. 또한 마크업 언어를 사용한 차트 표현 애플리케이션이므로 웹 브라우저에서 빠르고 선명하게 디스플레이할 수 있다. 본 논문에서 정의한 XML 기반의 차트 표현은 어느 플랫폼에서도 호환 가능하다.

본 논문에서 구현한 차트 편집 및 디스플레이 시스템을 사용하여 다양한 정보를 시각적으로 표현할 수 있다. 각 기관에서 실시한 설문조사의 결과를 표현하거나 교육기관의 통계 자료 그리고 학교에서 실시한 시험결과에 대한 분포 자료를 차트로 쉽게 표현할 수 있다.

향후 완전한 차트의 표현을 위해서는 표현 가능한 차트 유형 범위의 확대가 필요하고 사용이 간편하고 다양한 시각적 효과를 얻을 수 있는 편집기에 대한 연구가 필요하다.

## 참 고 문 헌

[1]"Microsoft Excel," http://www.microsoft.com/korea/Office/Excel

[2]"MATLAB 5.3.1 Introduction," http://www.mathworks.com/products/matlab/

[3]Brian Mathews, "Vector Markup Language(VML)," http://www.w3.org/TR/1998/NOTE-VML-19980513,1998.

[4]Nabeel Al-Shamma, "Precision Graphics Markup Language(PGML),"http://www.w3.org/TR/NOTE-PGML-19980410, 1998.

[5]한국전자통신연구원, "웹드와이드웹 컨소시엄 SVG의 작업 도안발표," http://etlars1.etri.re.kr/Bridfing/data/990227-2.DHTML, 1999.

[6]"VML(Vector Markup Language),"http://ce.kyungil.ac.kr/inguide/vml.html,1998.

[7]Lauren Wood and Vidur Apparao, "Document Object Model(DOM) Level 1 specification," http://www.w3.org/TR-REC DOM Level-1, 1999.