Journal of Digital Contents Society Vol. 9 No. 1 March. 2008(pp. 157-164)

XML을 이용한 차트 생성기

윤현님*, 김형일**

요 약

인터넷의 사용이 증가하면서 웹을 통해 정보를 표현하고 제공하는 수단으로서 차트의 활용도가 높아지고 있다. 차트는 데이터를 시각적으로 표현하므로 정보의 전달력이 높은 반면 차트 작성 프로그램 간의 호환성, 차트 데이터 공유의 어려움, 웹상에서 차트 출력의 문제점 등을 안고 있다. 본 논문에서는 차트 표현의 호환성과 차트 정보의 공유 문제를 해결하기 위해 XML을 활용하여 차트 정보를 표현하는 마크업 언어인 CML(Chart Markup Language)을 제안한다. CML은 웹 표준인 XML을 기반으로 하므로 프로그램 및 플랫폼 간의 호환성을 제공뿐만 아니라 차트 데이터의 공유가 가능하다.

A Chart Generator using XML

Hyunnim Yun*, Hyungil Kim**

Abstract

As use of internet gets increased, the efficiency of applying the chart which is expressing and providing information through web is increased. The chart has the high delivering ability of information because it expresses data visually. But there are problems such as compatibility between the chart composing programs, difficulty of sharing the chart data, and displaying the chart on the web. In this paper, we propose the chart markup language, which expresses the chart information by using the XML in order to solve problems of compatibility of the chart expression and sharing the chart information. The CML is possible not only to provide compatibility between platforms and programs but also to share the chart data as it is based on the XML.

Keywords : Chart Generator, XML, VML, DOM

1. 서론

정보의 시각적 표현 수단으로 차트를 많이 활용하고 있다. 그러나 웹상에서 차트를 사용하여 정보를 표현하기는 매우 어렵다[1][2]. 현재 차트의 작성에 많이 사용되고 있는 대표적인 프로그램에는 마이크로소프트사의 엑셀과 MathWorks사의 MATLAB이 있다[3][4]. 엑셀은 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 이것들은 서로 호환되지 않는 구조로 이루어져 있으며, 웹상에서 구현하기 어렵다는 문제점을 갖고 있다[5][6].

현재 웹상에서 차트를 작성하거나 디스플레이할 수 있는 시스템은 알려져 있지 않다. 반면 웹상에서 그래픽 처리를 위하여 개발된 XML 기반의 그래픽 마크업 언어들을 이용하여 차트를 그릴 수는 있다. 이러한 그래픽을 표현하는 XML 기반 응용으로는 VML, PGML, SVG가 있다. VML(Vector Markup Language)은 웹브라우저에서 그래픽을 구현할 수 있도록 W3C에서 제안한 XML 기반의 그래픽 마크업 언어이다[7]. HTML이 텍스트 정보의 마크업을 지원하는 것처럼 VML은 벡터 그래픽 정보의 마크업을 지원하며, 선과 곡선을 사용함으로써 그래픽을 디자인한다. VML은 웹상에서 구조적 데이터 표현을 위한 텍스트 기반 언어인 XML을 기반으로

※ 제일저자(First Author) : 윤현님
접수일자:2008년02월05일, 심사완료:2008년02월20일
* 한국폴리텍여자대학 디지털정보과 교수
yhnim@kopo.ac.kr
** 나사렛대학교 멀티미디어학과 교수

하므로 XML의 특징인 개방성, 단순성, 확장성을 지닌다[8][9]. 특히 XML의 단순하고 유연한 특성으로 인해 개발 및 편집이 쉬우며, 다양한 응용 영역 사이의 교환이 용이하다[10]. 또한 VML은 HTML 페이지 안에 삽입되어 전달되므로 외부 파일을 참조하는 방식보다 더 빠르게 그래픽을 처리할 수 있다. 이러한 VML은 CSS 2.0 뿐만 아니라 DOM(Document Object Model)과 같은 다른 W3C 표준을 지원한다[11]. 그러나 VML은 그래픽 요소를 표현하는데 필요한 속성의 사용이 복잡하고 어렵다는 단점이 있다.

PGML(Precision Graphics Markup Language)은 Adobe사의 포스트스크립트 및 PDF(portable document format)형식을 확장한 것으로 포스트스크립트 명령을 XML 요소와 속성으로 변환한 것이다[12]. 이러한 PGML은 특별한 플러그인이나 뷰어가 없어도 뛰어난 품질의 웹 기반 그래픽을 제공한다. PGML은 작지만 대부분의 그래픽에 대한 요구를 만족시킬 수 있는 강력한 기능을 가진 언어이다. PGML을 이용해 그린 그래픽은 하나 이상의 그래픽 객체를 모아 구성된다. 특히 PGML은 웹으로의 확장을 위해 간단한 애니메이션 컴포넌트와 복잡한 애니메이션의 동적인 처리를 위한 객체모델을 포함하고 있다. 그리고 PGML은 기존의 웹 표준화 규격인 HTML, XML, XSL, CSS, DOM, RDF과 같은 정의 규칙에 호환 가능하게 하였다. 그러나 PGML은 이진 파일 형식이므로 편집하기가 사실상 불가능하다는 점이 취약점으로 지적되고 있다.

비트맵에 기반을 둔 SVG(Scalable Vector Graphic)은 웹 그래픽 표현을 위해 많은 장점들을 제공한다[13][14][15]. SVG는 서로 다른 데이터베이스간의 그래픽 표현 데이터를 전달하는 기능, LAN 또는 웹 애플리케이션을 위해 관련 그래픽 및 텍스트를 간략하게 결합하는 기능, 다른 데이터 타입간의 변환 기능 등이 있다[16]. 반면 이러한 특징을 가진 SVG는 그래픽 개체의 형식을 지원하는 뷰어가 필요하므로 범용 웹브라우저에서 출력되지 않는다는 단점이 있다.

그래픽 처리를 위한 마크업 언어들은 XML을 기반으로 하였으므로 웹상에서 차트를 구현할 수는 있다. 그러나 다양한 그래픽에 대한 많은 표현 요소와 속성을 갖고 있기 때문에 그래픽 처리와 적용범위가 넓은 반면 차트를 구현하기에는 복잡하고 필요한 요소를 분류해내기가 어렵다. 특히 차트는 웹페이지 제작자나 웹 디자이너도 자주 활용하는 요소인데 반해 그래픽 마크업 언어처럼 사용법이 복잡하면 활용도가 떨어지는 문제가 발생한다. 그러므로 차트의 표현을 위한 필수적인 요소와 간단한 속성만으로 구성된 XML 기반 차트 표현 마크업 언어가 필요하다.

본 논문에서는 기존 차트 작성 프로그램과 그래픽 처리를 위한 XML 기반 마크업 언어의 장단점을 고려하여 편리하게 차트 정보를 표현하는 XML 기반의 차트 마크업 언어인 CML(Chart Markup Language)을 제안한다. 차트를 작성하기 위한 목적의 CML은 다른 XML 기반의 그래픽 응용 언어보다 효율성과 편의성 면에서 뛰어나므로 개발자뿐만 아니라 웹디자이너도 쉽게 활용할 수 있다. 또한 별도의 차트 작성을 위한 프로그램이 필요하지 않다는 범용성을 지니며 웹 표준에 근거하므로 플랫폼에 관계없이 웹브라우저에서 표현할 수 있다. 본 논문에서 제안한 CML은 XML 기반의 마크업 언어로 차트를 작성하므로 이진 데이터의 차트와는 달리 차트 데이터의 변환, 수정, 공유가 용이하며, 별도의 프로그램 없이 웹브라우저에서 차트를 출력할 수 있는 장점이 있다.

2. CML의 정의

XML을 이용한 차트의 표현을 위해서는 차트 작성에 필요한 항목들을 분석해야 한다. 그리고 분석 내용을 기반으로 CML의 DTD(Document Type Definition)를 정의해야 한다. CML의 DTD를 정의할 때는 기존의 XML 기반 그래픽 처리 마크업 언어의 복잡성을 탈피하여 차트의 구현에 필수적인 요소만으로 정의함으로써 차트의 구현이 쉽고 간단해야 한다[17]. 본 논문에서 제안한 CML에서는 차트 작성이 쉽고 편리할 수 있도록 CML DTD를 정의하였다.

2.1 CML DTD

DTD는 XML 문서의 내용 구조 모델을 설명하는 것으로 그 요소의 속성과 서로의 관계를

구조화하는 방법에 대한 정의이다. 그러므로 CML의 DTD를 작성하기 위해서는 먼저 차트 작성에 필요한 요소들에 대하여 분석하고 정의해야 한다. <표 1>은 CML DTD에서 표현할 주요 차트의 정보를 정리한 것이다.

CML은 차트의 정보를 마크업하기 위한 다양한 논리적 정보를 포함하고 있으며, 기존 HTML 사용자들도 쉽게 사용할 수 있을 정도의 간결한 구조를 제공한다. <표 1>의 내용을 토대로 CML DTD에는 요소들이 정의된다.

<표 1> CML DTD에서 표현하는 주요 차트 정보

요소	속성	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	차트의 제목
xtitle	position	차트의 항목 축 제목
ytitle	position	차트의 값 축 제목
axisinfo	없음	차트의 축 표현 정보
ymax	없음	차트의 값 축의 최대값
ymin	없음	차트의 값 축의 최소값
xaxis	없음	항목 축의 표시 정보
item1	color	각 항목 축
item2	color	각 항목 축
item3	color	각 항목 축
item4	color	각 항목 축
item5	color	각 항목 축
legend	없음	범례 정보
legend1	color	각 항목에 대한 범례
legend2	color	각 항목에 대한 범례
legend3	color	각 항목에 대한 범례
legend4	color	각 항목에 대한 범례
legend5	color	각 항목에 대한 범례
value	없음	데이터 정보
value1	color	각 항목의 데이터 값
value2	color	각 항목의 데이터 값
value3	color	각 항목의 데이터 값
value4	color	각 항목의 데이터 값
value5	color	각 항목의 데이터 값

<cml> 요소는 CML 문서의 루트 요소로 CML 문서의 시작을 알린다. <chart> 요소는 차트를 표현하기 위해 필요한 요소들을 포함한다. <charttype> 요소는 차트의 유형을 표현하며 “shape” 속성과 “planetype” 속성을 갖는다. 차트의 모양을 나타내는 “shape” 속성에는 원형(circle), 수평 막대(hbar), 수직 막대(vbar), 선형(line)과 같은 정보를 포함한다. “planetype” 속성은 차트의 바탕 영역에 대한 설정으로 “1”은 눈금선을 표시하며 “2”는 선을 표시하지 않는다. <chartinfo> 요소는 하나의 차트를 표현하는데 필요한 모든 요소를 포함한다. 포함하는 요소는 <header>, <yunit>, <title>, <axisinfo>, <value>, <legend> 요소이다.

<header> 요소는 차트의 제목을 표현한다. 차트 제목의 위치를 속성으로 갖는다. <yunit> 요소는 각 차트에서 나타내고 있는 데이터의 단위를 시각적인 효과를 위해 표현한다. <title> 요소는 차트의 항목 축(x축)의 제목과 값 축(y축)의 제목을 표현하기 위한 <xtitle>와 <ytitle> 요소를 포함한다. <xtitle>와 <ytitle>는 제목의 위치를 속성으로 갖는다.

<axisinfo> 요소는 <ymax>와 <ymin> 그리고 <xaxis> 요소를 포함하고 있다. <ymax> 요소는 값 축의 최대값을 표현하며, <ymin> 요소는 값 축의 최소값을 표현한다. <xaxis> 요소는 막대 차트 항목 축(x축)의 각 항목과 원 차트의 각 항목을 표현하기 위한 요소를 포함한다. <item1> ~ <item5>는 각 항목을 나타낸다.

<legend> 요소는 범례의 정보를 표현한다. 항목에 대한 범례 표현 값과 색깔에 대한 속성을 갖고 있다. 범례도 다섯 개의 항목을 가지며 <legend1> ~ <legend5>로 표현된다. 마지막으로 <value> 요소는 각 항목에 대한 막대의 높이와 원 차트를 표현하는데 사용되는 데이터를 표현한다. 또한 시각적 효과를 위해 색깔 속성을 포함한다. 항목 수에 따라 <valuel>, <value2>, <value3>, <value4>, <values> 요소를 포함한다. (그림 1)은 <표 1>의 차트 정보를 기반으로 작성한 CML DTD의 일부이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Attribute Definitions-->
<!--shape of charts-->
<!ENTITY % att-preinfo 'shape (circle|hbar|vbar)
        #IMPLIED'>
<!ENTITY % att-plane 'planetype (1|2) #IMPLIED'>
<!--position of axis's title-->
<!ENTITY % att-position 'position (center)
        #IMPLIED'>
<!ENTITY % att-color 'color CDATA #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)>
<!ELEMENT header (#PCDATA)>
<!ATTLIST header
 headerposition (up | down | right | left) #IMPLIED>
                    ⋮
```

(그림 1) CML DTD의 일부

2.2 CML 문서의 작성

차트의 정보를 표현하기 위해서는 CML 문서 인스턴스를 작성해야 하며, CML 문서는 CML DTD를 따른다. <표 2>는 CML 문서 인스턴스로 작성할 연간 가전제품 판매량에 대한 예제이다.

<표 2> 연간 가전제품 판매량에 대한 데이터

	TV	냉장고	오디오	캠코더	컴퓨터
1/4 분기	30	30	35	35	60
2/4 분기	35	40	40	50	40
2/4 분기	35	30	35	45	50
3/4 분기	20	20	30	30	30
총 판매량	120	120	140	160	180
평균 판매량	**30**	**30**	**35**	**40**	**45**

(그림 2)는 <표 2>의 각 가전제품에 대한 연평균 판매량을 막대 차트로 표현하기 위하여 작성한 CML 문서 인스턴스이다. CML 문서에서 각 가전제품은 “item” 과 “legend” 요소로 이용되며, 평균 판매량은 “value” 요소에 사용된다. (그림 2)에서 ①은 차트의 종류와 차트를 그리기 위한 기본영역의 유형을 표시하는 부분이다. 예제에서는 “shape=vbar”이므로 차트는 수직 막대이고, “planetype”이 “1”이므로 막대 차트의 기본 영역은 눈금선으로 나타난다. ②는 차트의 제목을 표현하는 부분으로 제목은 “Home Appliance” 이다.

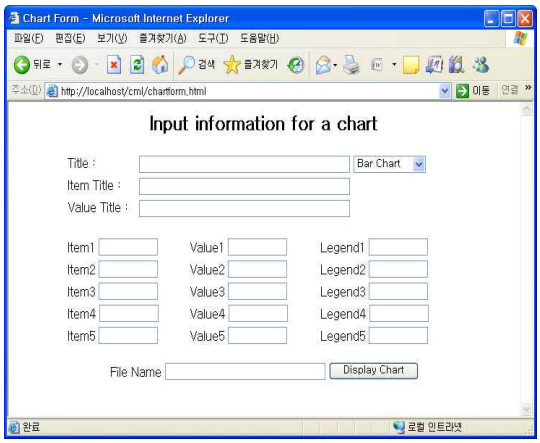
```
<CML>
  <chart>
    <charttype shape="vbar" planetype="1"/> ---①
    <chartinfo>
      <header headerposition="up">Home Appliance --②
      </header>
      <title> ---③
        <xtitle position="center">Item</xtitle>
        <ytitle position="center">Average</ytitle>
      </title>
      <axisinfo>
        <ymax>50</ymax> ---④
        <ymin>0</ymin>   ---⑤
        <xaxis> ---⑥
          <item1>TV</item1>
          <item2>Ref</item2>
          <item3>Aud</item3>
          <item4>Cam</item4>
          <item5>Com</item5>
        </xaxis>
      </axisinfo>
      <value> ---⑦
        <value1 color="#FFFFCC">30</value1>
        <value2 color="#FFCCFF">30</value2>
        <value3 color="#CCFFFF">35</value3>
        <value4 color="#CCCCFF">40</value4>
        <value5 color="#CCCCCC">45</value5>
      </value>
      <legend> ---⑧
        <legend1 color="#FFFFCC">TV</legend1>
        <legend2 color="#FFCCFF">Refregerator
        </legend2>
        <legend3 color="#CCFFFF">Audio</legend3>
        <legend4 color="#CCCCFF">Camcoder</legend4>
        <legend5 color="#CCCCCC">Computer</legend5>
      </legend>
    </chartinfo>
  </chart>
</CML>
```

(그림 2) 가전제품 연평균 판매량에 대한 CML 문서의 예

③은 막대 차트의 x축 제목과 y축 제목을 작성하는 부분이다. ④와 ⑤는 값 축의 최대값과 최소값을 표현하는 부분이다. 최대값과 최소값은 막대 차트로 표현하고자 하는 데이터의 범위를

미리 파악하여 차트 작성자가 작성한다. 예제에서는 최대값을 50, 최소값을 0으로 설정하였다. ⑥은 각 항목을 표현하는 부분으로 항목 축에 표현될 내용이다. ⑦은 각 항목의 데이터 값들을 작성하는 부분으로 실제 차트에서 그래픽으로 표현될 수치들이 작성되는 부분이다. 수직 막대 차트에서는 “value1”부터 “value5”에 입력한 다섯 개의 데이터들을 막대의 높이를 표현하는데 사용한다. ⑧은 범례를 표현하는 부분으로, 막대의 색깔과 항목에 대한 정보를 표현한다.

본 논문에서 제안한 CML에서는 간단하고 쉽게 CML 문서를 작성할 수 있도록 CML 문서 작성기를 설계하여 활용하였다. 문서 작성기에 입력하는 정보로는 차트의 제목, 차트의 각 축의 제목, 각 항목 이름, 데이터 값 그리고 범례가 있다. 사용자가 입력한 정보는 (그림 2)와 같은 CML 문서로 저장된다. (그림 3)에 CML 문서 작성기 화면을 나타내었다.



(그림 3) CML 문서 작성기 화면

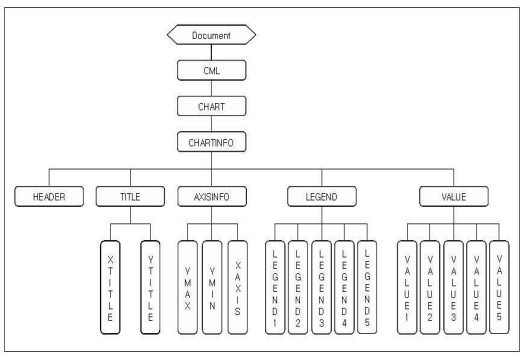
3. CML 문서의 출력

CML로 작성된 문서는 웹 표준을 따르고 있으므로 특정한 프로그램이 필요하지 않다. 그러므로 본 논문에서는 CML 문서를 출력하기 위하여 웹브라우저를 사용한다. 본 논문은 윈도우즈 XP 운영체제 환경에서 CML 문서를 웹브라우저로 출력하였으며, 그래프의 출력을 위하여 마이크로소프트사의 XML 파서, ECMA DOM 스크립트, 인터넷 익스플로러 6.0을 이용하였다.

3.1 CML 문서의 변환

CML 문서의 차트 표현 정보를 출력하기 위해서는 CML 문서를 그래픽 표현 언어인 VML 문서로 변환하는 과정이 필요하다. VML은 기존 비트맵 방식과 달리 곡선과 직선을 수학적 알고리즘으로 표현하므로 그래픽 구성이 훨씬 간단해 파일 크기와 출력 속도를 기존 비트맵 방식보다 줄일 수 있다는 장점이 있다.

CML 문서를 VML 문서로 변환하는 과정에는 DOM을 이용한다. DOM은 HTML과 XML 문서를 다루기 위한 트리 기반 API 표준으로 플랫폼 및 언어 중립적인 인터페이스이다[18][19]. DOM을 이용하면 XML 문서를 객체화하여 이를 DOM 트리 형태로 접근할 수 있다[20]. CML 문서의 차트 표현 정보를 출력하기 위해서는 DOM을 이용하여 변환하는 과정이 필요하다. CML 문서를 차트로 출력하기 위해서는 DOM 인터페이스를 통해 CML 문서를 DOM 트리로 생성한 후, 변환 모듈을 통해 DOM 트리에서 차트에 해당하는 정보들을 탐색하여 VML 코드로 변환한다. (그림 4)는 CML 문서를 DOM 스크립트를 통해 DOM 트리로 변환한 결과이다.



(그림 4) CML 문서에 대한 DOM 트리

(그림 5)는 (그림 2)의 CML 문서에서 막대 차트의 각 항목에 대한 데이터 값을 작성한 ⑦ 부분을 VML로 변환하기 위한 DOM 스크립트 코드이다. 먼저 요소가 “value”인지 확인한 후 “value” 노드의 자식 노드인 “value1”의 값을 CML 문서로부터 읽는다. 다음으로 웹브라우저에 출력할 막대 차트의 위치를 계산하여 VML 요소로 표현한다. 그 다음에는 “value1”과 동일한 방법으로 “value2”에서 “value5”까지 막대 차

트를 작성한다. 이 DOM 스크립트 코드는 자바 스크립트를 이용하여 작성하였다.

```
// VALUE
else if( ciElem.nodeType==1 &&
        ciElem.nodeName=="value"){
 var T = 0;
 var L = 50;
 for(k=0; k<ciElem.childNodes.length; k++){
   var value = ciElem.childNodes(k);
   var A;
   if( value.nodeType==1 &&
       value.nodeName=="value1"){
        A = value.firstChild.nodeValue;
   }else if(value.nodeType==1 &&
            value.nodeName=="value2"){
        A = value.firstChild.nodeValue;
   }else if(value.nodeType==1 &&
            value.nodeName=="value3"){
        A = value.firstChild.nodeValue;
   }else if(value.nodeType==1 &&
            value.nodeName=="value4"){
        A = value.firstChild.nodeValue;
   }else if(value.nodeType==1 &&
            value.nodeName=="value5"){
        A = value.firstChild.nodeValue;
   }else{
   }
   T = 300-A;
   L += 50;
   document.write( "<v:rect fillcolor='"
      +value.getAttribute('color')
      +"'style='position:absolute; top:"+T+"px; left:"
      + L + "px; width:40px; height:" +A+"px'>\n");
   document.write( "</v:rect>\n");
 }
}
                    ⋮
```

(그림 5) CML 문서의 Value 요소를 VML로 변환하기 위한 DOM 스크립트

(그림 6)은 앞에서 설명한 과정을 거쳐 변환된 VML 문서의 일부이다. (그림 6)에서 “<v:>”는 VML임을 알리는 요소이다. “v:” 다음에 나오는 내용은 브라우저에 출력할 그래픽의 유형을 나타내는 것으로 선을 의미하는 “line”, 사각형을 의미하는 “rect”, 원을 의미하는 “oval” 등이 존재한다. 그리고 VML 요소 내에는 색상을 지정하는 “fillcolor” 속성, 웹브라우저에서 그래픽을 출력할 위치를 지정하는 “style” 속성 등이 있다. 예를 들면, (그림 6)의 첫 번째 내용은 웹브라우저의 왼쪽 상단 부분에서 오른쪽으로 100 픽셀, 아래쪽으로 270 픽셀 이동한 점에서 너비 40픽셀, 높이 30픽셀의 사각형을 그리고 색상을 '#FFFFCC'로 채우는 것이다.

```
<v:rect fillcolor='#FFFFCC' style='position:absolute;
 top:270px; left:100px; width:40px; height:30px'>
</v:rect>
<v:rect fillcolor='#FFCCFF' style='position:absolute;
 top:270px; left:150px; width:40px; height:30px'>
</v:rect>
<v:rect fillcolor='#CCFFFF' style='position:absolute;
 top:265px; left:200px; width:40px; height:35px'>
</v:rect>
<v:rect fillcolor='#CCCCFF' style='position:absolute;
 top:260px; left:250px; width:40px; height:40px'>
</v:rect>
<v:rect fillcolor='#CCCCCC' style='position:absolute;
 top:255px; left:300px; width:40px; height:45px'>
</v:rect>
                    ⋮
```

(그림 6) VML로 변환된 Value 요소 부분

3.2 레이어를 이용한 CML 문서의 출력

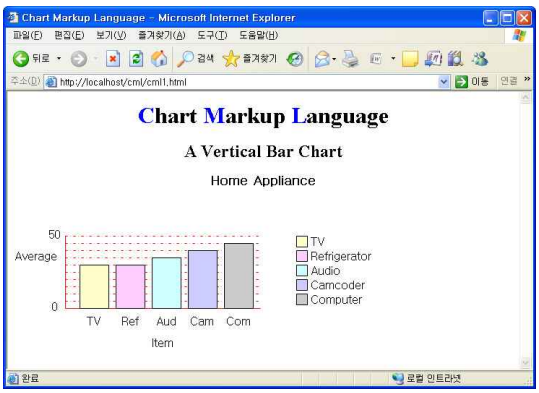
본 논문에서는 CML 문서를 VML 문서로 변환하는 동시에 DHTML의 레이어 기능을 이용하여 차트의 요소들을 웹브라우저에 출력한다. DHTML은 기존 HTML에 비해 애니메이션 기능과 사용자 상호작용 기능을 강화한 것으로 새로운 HTML 태그, 옵션, 스타일 시트 등을 의미한다. 그리고 레이어는 웹페이지 내에 작성된 내용 영역에 관계없이 자유로이 영역을 지정하여 내용을 겹쳐서 출력할 수 있는 기능이다. 레이어를 이용하기 위해서는 “style” 속성에서 내용을 출력할 “position” 값을 절대 위치인 “absolute”로 설정하고 내용의 영역을 “top”, “left”, “width”, “height”을 이용하여 지정한다. DHTML의 레이어를 이용하여 출력하였을 경우에 차트의 출력 시간을 단축시키고 출력 문서의 모양을 정형화할 수 있다는 장점을 갖는다. (그림 7)은 차트의 기본 영역을 표현하기 위해 레이어를 사용한 예의 일부이다.

본 논문에서는 제안한 CML을 이용하여 작성된 CML 문서를 출력하기 위하여 웹브라우저를 사용하였다. 웹브라우저를 사용함으로써 별도의 출력 애플리케이션 없이도 사용자가 간단하게

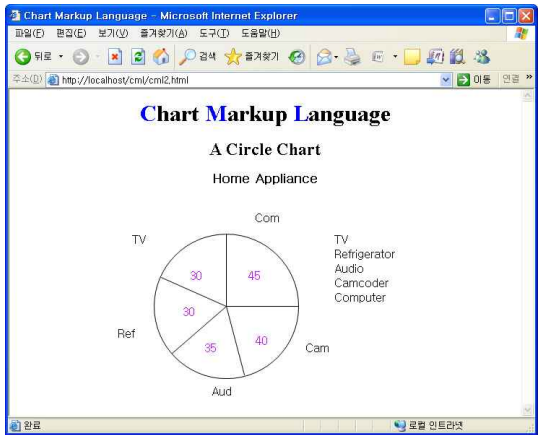
차트를 디스플레이 할 수 있다. 또한 인터넷 익스플로러 5.0 이상에서는 DOM API를 지원하는 프로세서가 내장되어 있기 때문에 CML 문서를 쉽게 변환할 수 있다. (그림 8)과 (그림 9)는 <표 2>의 차트 데이터를 이용하여 작성된 막대형, 원형 차트의 CML 문서를 웹브라우저에 각각 출력한 결과이다.

```
function paint_bar_chart_basis(chart_basis_first_position){
  chart_basis_first_position = 200;
  document.write("<v:line style='position:absolute;
   top:0px; left:0px;' from='80px, 300px'
   to='350px,300px' strokecolor=red strokeweight=0.5pt>
   \n");
  document.write("<v:stroke dashstyle=solid />\n");
  document.write("</v:line>\n");
  document.write("<v:line style='position:absolute;
  top:0px; left:0px;' from='80px, 300px'
  to='80px,200px' strokecolor=red strokeweight=0.5pt>\n");
  document.write("<v:stroke dashstyle=solid />\n");
  document.write("</v:line>\n");
  var basePos=200;
  for(i=0; i<10; i++){
   document.write("<v:line style='position:absolute;
    top:0px; left:0px;' from='80px, "+basePos+"px'
    to='350px,"+basePos+"px' strokecolor=red
    strokeweight=0.5pt>\n");
   document.write("<v:stroke dashstyle=dashdot />\n");
   document.write("</v:line>\n");
   basePos +=10;
  }
}
                    ⋮
```

(그림 7) 레이어를 사용한 예의 일부



(그림 8) 막대형 차트로 출력



(그림 9) 원형 차트로 출력

4. 결론 및 향후 연구

본 논문에서는 차트의 표현 및 디스플레이를 위한 기존의 차트 정보를 표현하는 방법들의 문제점들을 해결하기 위해 CML을 제안한다. CML을 이용할 경우 XML 기반의 그래픽 마크업 언어보다 사용이 편리하므로 웹페이지 제작자나 웹디자이너도 손쉽게 활용할 수 있다. 또한 CML을 이용할 경우 범용 웹브라우저를 통해 차트의 정보를 전달할 수 있으며, 별도의 프로그램이나 호환기 없이도 웹브라우저에 차트 출력이 가능하다. 특히 XML 기반의 마크업 언어로 차트를 작성하므로 차트 데이터의 변환, 수정, 공유가 용이하다. 본 논문에서 정의한 CML은 XML을 기반으로 하므로 어느 플랫폼에서도 호환 가능하며 DOM을 이용하여 문서 변형이 자유롭다.

본 논문에서 제안한 CML을 이용하면 다양한 정보를 웹브라우저 상에서 시각적으로 표현할 수 있다는 장점이 있기 때문에, 이러한 장점을 활용하면 각 기관에서 실시한 설문조사의 결과나 교육기관의 통계자료 그리고 학교에서 실시한 시험결과를 이해하기 쉬운 시각적인 도구로 쉽게 표현할 수 있다.

향후 과제로는 다양한 정보가 입력되더라도 효율적인 시각 자료를 생성할 수 있도록 범용적 차트 생성기에 대한 연구와 문서 정보에 적합한 시각화된 자료가 자동 생성될 수 있는 자동화 연구가 필요하다.

참 고 문 헌

[1] P. L. Thomas, D. F. Brailsford, “Enhancing composite digital documents using XML-based standoff markup,” Proceedings of the 2005 ACM symposium on Document engineering, 2005.

[2] S. Ronnau, J. Scheffczyk and U. M. Borghoff, “Towards XML version control of office documents,” Proceedings of the 2005 ACM symposium on Document engineering, 2005.

[3] C. Stinson, M. Dodge, Microsoft Office Excel 2003 : Inside Out, Microsoft Press, 2003.

[4] D. C. Hanselman, B. L. Littlefield, Mastering MATLAB 7, Prentice Hall, 2004.

[5] S. Abiteboul, P. Buneman and D. Suciu, “Data on the Web,” Morgan Kaufmann Publishers, 2000.

[6] F. Bes, C. Roisin, “A presentation language for controlling the formatting process in multimedia presentations,” Proceedings of the 2002 ACM symposium on Document engineering, 2002.

[7] B. Mathews, “Vector Markup Language(VML),” http://www.w3.org/TR/1998/NOTE-VML-19980513, 1998.

[8] C. F. Goldfarb and P. Prescod, The XML Handbook, Prentice Hall, 1998.

[9] A. Zisman, "An Overview of XML," Journal of Computing and Control Engineering, 2000.

[10] S. Holxner, XML Complete, McGraw-Hill, 1998.

[11] I. Zaslavsky, “A New Technology for Interactive Online Mapping with Vector Markup and XML,” Cartographic Perspectives, 2000.

[12] N. Al-Shamma, “Precision Graphics Markup Language(PGML),” http://www.w3.org/TR/NOTE-PGML-19980410, 1998.

[13] P. Topping, “Mathematics on the Web : MathML and MathType,” http://www.dessci.com/en/reference/white_papers/mt_mathml.htm, 1999.

[14] J. Ferraiolo, “Scalable Vector Graphics(SVG) Specification,” http://www.w3.org/TR/1999/WD-SVG-19990211/, 1999.

[15] R. H. Landau, D. Vediner, P. Wattanakasiwich and K. R. Kyle, "Future Scientific Digital Documents with MathML, XML, and SVG," Computing in Science and Engineering, 2002.

[16] J. C. Mong, D. F. Brailsford, “Using SVG as the rendering model for structured and graphically complex web material,” Proceedings of the 2003 ACM symposium on Document engineering, 2003.

[17] G. J. Bex, F. Neven and J. V. Bussche, “DTDs versus XML schema: a practical study," Proceedings of the 7th International Workshop on the Web and Databases, 2004.

[18] L. Wood and V. Apparao, “Document Object Model (DOM) Level 1 specification,” http://www.w3.org/TR-REC DOM Level-1, 1999.

[19] A. L. Hors, P. L. Hegaret, L. Wood, G. Licol, J. Robie, M. Champion, S. Byrne, “Document Object Model (DOM) Level 3 Core Specification,” http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/, 2004.

[20] S. Widergren, A. deVos, and Z. Jun, “XML for Data Exchange,” Power Engineering Society Summer Meeting, IEEE, 1999.

윤 현 님

2000년 동국대학교 컴퓨터교육학과(교육학석사)

2003년 동국대학교 정보통신공학과(박사수료)

1996년～1999년 : (주)유니파이코리아

2001년～현 재 : 한국폴리텍여자대학 디지털정보과 교수

관심분야 : XML, e-Learning, 그리드, 분산 및 네트워크 컴퓨팅, 게임

김 형 일

2001년 :동국대학교 컴퓨터공학과(공학석사)

2004년 :동국대학교 컴퓨터공학과(공학박사)

1996년～1998년 : (주)경기은행

2005년～2006년 : 동국대학교 컴퓨터공학과 IT교수(정보통신부)

2007년～현 재 : 나사렛대학교 멀티미디어학과 교수

관심분야 : 지능형시스템, 추천시스템, 기계학습, 게임,