

# MathML에 기반한 수학적 편집 및 표현 시스템

강치원<sup>†</sup> · 전성학<sup>††</sup> · 정희경<sup>†††</sup>

## 요 약

본 논문은 MathML에 기반하여 수학적식을 웹상에서 표현하기 위한 수학적 편집 및 표현 시스템의 설계 및 구현에 관한 것이다. 본 시스템은 W3C에서 제안된 MathML의 제안을 준수하여 수학적식을 보다 쉽고 효율적으로 편집할 수 있도록 사용자 중심의 다양한 인터페이스를 제공하도록 설계되었다. 편집시스템에서 작성된 수학적식은 표현시스템을 통하여 바로 결과를 확인할 수 있도록 구현하였다. 또한 데이터를 처리하는 속도를 향상시키기 위해 기존 XML파서의 기능을 보완하여 본 시스템에 맞도록 설계하고 구현하였다. 본 편집 및 표현 시스템은 웹상에서의 수학적식 사용의 보편화에 기여하고, XML에 기반하여 수학적식을 포함하는 문서의 효율적인 처리 등에 유용하게 사용될 것이다.

## Mathematics Equation Editing and Presentation System based on MathML

Chi-won Kang<sup>†</sup> · Sung-hak Jun<sup>††</sup> · Hoe-kyung Jung<sup>†††</sup>

## ABSTRACT

In this paper, we present an equation edition system, which is designed and implemented according to the W3C MathML standard. The system can be used to edit equations for the various web-oriented applications. Various user-friendly interfaces are provided for the easy and efficient edition of equations. The edited equations can be viewed immediately with the representation modules. To improve the processing speed of equations, we upgraded functions of the conventional XML parser. The presented system will contribute to the extensive usage of equations on the web and efficient processing of XML documents with equations.

**키워드:** XML, XSL, MathML, XML 파서(XML Parser), 수학적식(Mathematics Equation), 편집 시스템(Editing System), 포매팅(Formatting), 문서처리(Document Processing)

### 1. 서 론

오늘날 인터넷이 생활 속에 자리잡은 것은 표현의 단순함과 보편성을 제공하는 웹의 역할이 크다 할 수 있다. 그러나 이러한 웹은 초기에 과학자들 사이의 정보 교환에 목적을 두고 만들어진 것임에도 불구하고, 과학에 관련된 정보를 표현하는데 있어 기본적으로 쓰이게 되는 수학적식을 웹상에서 표현하는 데에는 많은 제약적인 요소가 존재하고 있다[3, 8]. 수학적식을 웹상에서 표현하는 것은 보통, 이미지와 텍스트를 섞어서 사용하는 것이 대부분이지만 이는 읽고 표현하기 어려울 뿐만 아니라, 이미지 특성상 상대적으로 큰 데이터 크기로 인해 처리와 교환에 부담을 주게 되고, 수학적식의 검색이나 인덱싱, 그리고 다른 수학과 관련된 어플리케이션에서의 재사용을 어렵게 한다[1, 2, 7]. 이러한 문제점들을 보완하기 위해 W3C(World Wide Web Consortium)에서는 웹상에서 수학적식 표현의 효율적인 표기 및 저장, 공

유를 가능하게 하기 위해 XML(eXtensible Markup Language) 기반의 수학적식 표기 언어인 MathML(Mathematical Markup Language)을 제정하였다[1, 2, 7]. MathML은 이러한 기존의 단점을 극복하고자 수학적 표현과 내용을 마크업 기반의 코드로 정의하였으며, 웹상에서 수학적식 표현과 처리에 효과적인 대안을 제시하고 있다. MathML을 사용하기 위해서는 MathML 문서의 작성과정이 필수적이다. 또한, 올바른 MathML의 사용을 위해서는 MathML의 구조 및 복잡한 MathML DTD(Document Type Definition)에 관련된 지식을 요구한다[3, 6].

이를 위해 본 논문에서는 MathML에 관한 기초 기술 연구 및 구조화된 MathML 문서를 사용자 중심의 편집 인터페이스를 통해 쉽게 편집가능한 편집기 및 포매팅하여 표현하기 위한 렌더링 엔진을 설계 및 구현한다. 따라서 MathML 문서를 편집하여 사용자가 요구하는 MathML 문서를 표현하거나 포매팅까지 포함하여 문서교환이 가능하다. 또한 수학적식을 포함하는 XML 문서처리 환경 구축에 크게 기여하고, 초고속정보망에서 수학적식을 포함하는 전자문서처리, 저장, 검색을 위한 기본 응용개발을 촉진시키고 멀티미디어 문서처리 시스템 기술 개발 촉진에 중요한 역할을 할 것이

\* 본 연구는 한국과학재단 목적기초연구(98-0102-03-01-3)지원으로 수행되었음.

† 준희원 : 배재대학교 대학원 컴퓨터공학과

†† 정희원 : OCI 정보통신 연구소 책임연구원

††† 정희원 : 배재대학교 컴퓨터공학과 교수

논문접수 : 2001년 12월 11일, 심사완료 : 2002년 2월 25일

다[1, 2, 3, 7].

## 2. MathML

### 2.1 MathML 개념 및 소개

XML 어플리케이션 중의 하나인 MathML은 전문적으로 수학적식을 표현하기 위해 설계되었고, 또한 메타언어로서 문서의 구조와 내용을 담을 수 있다. MathML의 최종목적은 수학적식이 웹을 통하여 표현되고 활용되는 것으로 HTML언어가 웹상에서 활용되는 것처럼 어렵고 복잡한 수학적식을 웹상에서 표현하는 것이다.

MathML은 수학적 표현과 수학적 내용을 인코딩할 수 있는 기반기술을 제공한다. MathML 태그 중의 약 30개는 표현 엘리먼트(Presentation Element)들로 추상적인 표기법의 구조를 기술하며, 또 다른 150개는 내용 엘리먼트(Contents Element)들로 수학적 표현에 담긴 수학적 의미를 정확하게 기술한다. MathML의 엘리먼트는 표현 마크업, 내용 마크업으로 구성된다[1, 2].

표현 마크업은 수학적식의 표현에 초점을 두고 주로 시각적으로 지향하는 2차원의 구조를 기술한다. 전형적인 예로, mrow 엘리먼트는 수학적식에서 수학적 표현의 하위 그룹을 묶어놓기 위하여 사용되고, msup 엘리먼트는 윗첨자를 표현한다.

표현 마크업은 30개의 엘리먼트와 그에 상응하는 50개 이상의 속성을 갖고 있다. 본 마크업에 포함되는 대부분의 엘리먼트는 레이아웃 스키마(Layout Schema)로서 다른 표현 엘리먼트를 포함 가능하다. 그 외에도 수학적식의 최소단위를 표현하기 위한 mi, mn, mo와 같은 토큰 엘리먼트들과 빈 엘리먼트가 있다. 레이아웃 스키마는 몇 가지 클래스로 분류할 수 있는데, 첫째 그룹은 스크립트를 표현하기 위한 것으로서 msub, munder, mmultscripts 등의 엘리먼트들을 포함한다. 둘째로는 더 보편적으로 쓰이는 레이아웃 스키마로서 mrow, mstyle, mfrac 등의 엘리먼트를 포함한다. 셋째 그룹은 테이블과 행렬에 관한 것이다. 그외에도 maction이라는 엘리먼트가 단독으로 존재한다[1, 2].

내용 마크업은 표현 마크업과는 달리 수학적식의 추상적인 내용을 표현하는 데 초점을 둔다. 예를 들어, 윗첨자를 표시할 때 표현 마크업에서처럼 위치를 나타내는 엘리먼트를 사용하는 것이 아니라 base와 exponent라는 두 연산자를 필요로 한다. 그렇기 때문에, 수학적 노테이션(Notation)의 배치는 근본적인 수학적 물체의 논리적인 구조를 밀접하게 따라간다.

그 외에도 표현 엘리먼트와 내용 엘리먼트를 혼합하여 사용하는 경우가 있다.

본 논문은 MathML을 통한 수학적식의 표현에 목적을 두고 있기 때문에 표현 마크업을 기반으로 하여 설계 및 구현하였다.

### 2.2 표현 마크업

표현 마크업은 소수의 특수한 경우를 제외하고는 모두

기초적인 부호와 기호를 나타내기 때문에 전통적인 수학적 표기법과 유사하다고 할 수 있다. 우리가 텍스트 문서의 문법구조를 타이틀, 구절, 단락 등으로 파악할 수 있는 것처럼 표현 엘리먼트들은 그 자체가 수학적 표기법의 문법구조를 담고 있다. 예를 들면,  $x+a/b$ 라는 수학적식이 있을 때 (그림 1)과 같이 mrow라는 엘리먼트를 여러 번 포함관계 를 이루어 사용한다[1, 7].

```
<math>
  <mrow>
    <mi>x</mi>
    <mo>+</mo>
    <mrow>
      <mi>a</mi>
      <mo>/</mo>
      <mi>b</mi>
    </mrow>
  </mrow>
</math>
```

(그림 1) 표현 마크업의 예

#### 2.2.1 표현 엘리먼트

표현 엘리먼트들은 다음과 같은 것들이 있다.

- 토큰 엘리먼트(Token Elements)
 

표현 마크업 언어에서 토큰 엘리먼트는 수학적 표기법에서 의미상의 제일 작은 단위를 표현한다. 즉 텍스트 문서에서 제일 작은 단위를 나타내는 단어에 비유할 수 있다. 토큰 엘리먼트들은 보통 한 개의 문자로 수학적 기호를 표시한다. 그러나 다른 경우도 있는데 예를 들면 함수들의 이름이거나 한 개 이상의 문자로 구성된 토큰들이다. 더 나아가서 전통적인 수학적 표기법에서 기호가 나타내는 의미는 문자의 크기, 스타일(예를 들면, 굵은 선으로 표시된 'X'는 벡터를 표시)등과 연관된다. 그렇기 때문에 문자, 토큰, 스타일은 MathML에서 서로 밀접히 연관되어 있다고 볼 수 있다.
- 범용 레이아웃 스키마타
 

범용 레이아웃 스키마타(General Layout Schemata)는 토큰 엘리먼트와 여러가지 스크립트 형태를 표시하는 엘리먼트, 그리고 테이블이나 행렬을 표시하는 엘리먼트를 제외한 나머지 엘리먼트들로, 가장 보편적으로 사용되는 엘리먼트이다. 범용 레이아웃 스키마타 엘리먼트에는 분수, 근호 및 보편적으로 쓰이는 함수 등이 포함된다.
- 스크립트와 스키마타 제한
 

스크립트 엘리먼트들은 하나의 베이스를 기준으로 하여 고정 위치를 표시한다. 수학적 상에서 기호의 고정된 위치에 스크립트들이 붙는 것은 아주 보편적이다.
- 테이블과 행렬
 

테이블이나 행렬은 mtable 엘리먼트를 사용하여 표현한다. mtable의 자식 엘리먼트로는 행(Row)을 정의하는 mtr이나 mlabeledtr 엘리먼트가 올 수 있으며, 그 하위로는 열(Column)을 정의하는 mtd 엘리먼트가 오게된다.

- **효과적인 표현**  
수학식을 효과적으로 표현하는 데는 여러가지 방법이 있다. 그 중, MathML의 서브 표현에 링크를 연결하는 것이 한 가지 방법인데, 이와 같은 액션(action)을 수학식 표현에 바인딩(binding)하기 위하여 MathML은 maction이라는 엘리먼트를 제공한다. maction 엘리먼트의 아래로 임의개수의 자식 엘리먼트가 올 수 있다.
- **속성**  
표현 엘리먼트들은 50여 개의 속성을 갖고 있다. MathML문법에서 속성은 XML에서 속성을 표현하는 것과 동일하다. 각자 엘리먼트가 가질 수 있는 속성들 외에도 CSS나 XSLT와 같은 스타일 시트를 사용할 수 있도록 하기 위해 모든 엘리먼트는 class, style과 id라는 속성을 공동으로 갖는다. 또한, MathML표현식에서 링크 기능을 사용할 수 있게 하기 위해 xlink:href라는 속성을 갖는다. 그 외에도 모든 엘리먼트에 올 수 있는 속성은 xref와 deprecated가 있다.

### 3. MathML 편집 시스템 설계

#### 3.1 전체 시스템 구성

(그림 2) 시스템 전체 구성도

(그림 2)는 본 편집 시스템의 전체 구성도이다. 입력되거나 초기화된 MathML문서는 문서 처리부의 파서를 통해 파싱 과정을 거쳐 DOM(Document Object Model)과 같은 형태인 MathML 노드 구조로 구성되며, 구조 관리를 통해 관리된다. MathML 노드 객체는 구조 관리를 통해 수학적 포매팅에 필요한 MathML 포매팅 노드 구조정보로 변환되어 디스플레이 처리부에 전달되고, 이를 토큰 데이터와 레이아웃 컴포넌트 데이터로 구분하여 처리하게 된다. 이 과정에서 각 엘리먼트들의 속성에서 정의된 스타일 정보가 스타일 처리기를 통해 적용되고 포매팅되어 MathML 뷰 영역에 브라우징 된다. 사용자는 이러한 정보와 문서 편집부에서 제공하는 편집 인터페이스를 통해 원하는 형태로의 MathML 문서 작성이 가능하게 된다. 사용자의 편집을 통해 변경되고 생성된 각각의 편집 인터페이스 데이터는 문서 처리부에 전달되어 MathML 노드 구조를 갱신하게

되고, 이는 다시 다른 편집 인터페이스 데이터와 포매팅 정보에 적용된다.

#### 3.2 문서 처리 인터페이스 설계

문서 처리부는 입출력을 담당하며, 입력된 문서를 검증하는 파서 부분과 파서를 거친 문서의 구조를 관리하는 MathML 구조관리기로 구성된다.

##### 3.2.1 XML 파서 설계

본 논문에서는 MathML문서를 파싱하여 문서에서 선언한 내용에 따라 원하는 형태의 결과물을 DOM 형태의 결과로 변환해 주기 위한 기존의 XML파서를 배제하고, 이를 직접 구현함으로써 속도 향상 및 외부 모듈로부터의 의존도를 최소화 하였다. 파싱 결과를 가지고 MathML문서의 실제 포매팅에 사용되는 MFOT(MathML 포매팅 객체 트리)의 자료 구조로 변환해 주는 모듈을 설계하고 구현하였다. (그림 3)은 XML관리를 통해 XML문서가 파싱되어 XML 노드구조가 생성되는 모습을 보인다.

(그림 3) 파싱을 통한 XML 노드의 생성과정

파싱하고자 하는 XML문서를 문자열의 형태로 토큰처리에 전달하게 되면 토큰처리는 이를 순차적으로 검색하여, PI(Processing Instruction)가 인식됐을 경우, 시작 태그가 인식됐을 경우, 그리고 빈태그와 종료태그가 인식됐을 경우를 각각 감지하여 각 경우에서 처리할 수 있는 데이터와 구조를 생성하게 된다. 파싱 시작시 문서의 최상위를 나타내는 Document 타입의 XML 노드를 생성하게 되며, 이어서 감지된 PI들은 PI Event를 통해 생성된다. 시작태그가 인식되는 경우에는 Element Start Event 처리를 통해 Element 타입의 XML 노드를 생성하게 되며, 현재 처리되고 있는 노드로 등록하게 된다. 시작태그 내에 속성이 검색되었을 경우에는 Attribute Event 처리를 통해 하나 이상의 속성 그룹을 해당 XML 노드 객체내의 속성정보 객체로 설정한다. 종료태그가 감지된 경우에는 해당 태그가 값을 포함하는 단말 엘리먼트의 구성 요소일 수 있다. 이런 경우, 시작태그와 해당 태그 사이에 있는 문자열은 엘리먼트의 값을 나타내는 것이므로, 이를 추출하여 Text Event를 통

해 해당 XML 노드에 텍스트 타입의 XML 노드를 추가하게 된다. 또한, 엘리먼트 종료 이벤트를 통해 엘리먼트 시작 이벤트 과정에서 등록된 현재 처리되는 노드 정보를 해당 부모 노드로 설정하고 반환하게 된다. 빈 태그가 인식되는 경우에는 엘리먼트의 값 및 종료 태그가 존재하지 않는 경우가 되기 때문에, 텍스트 이벤트가 발생하지 않게 되며, 종료태그로서의 역할을 하기 때문에 엘리먼트 시작 이벤트 및 속성 이벤트, 엘리먼트 종료 이벤트를 모두 처리하게 된다. 이렇게 태그의 종류를 분별하고 이를 통해 처리할 수 있는 이벤트들을 수행하여, 입력된 XML문서를 XML 노드들로 구성된 DOM형태로 구조화 할 수 있는 것이다[5]. 이 구조화 정보를 통해 문서 처리부에서는 본 편집 시스템의 편집 인터페이스를 통해 생성되고 변경되는 데이터 및 MathML문서가 포맷팅 되는 정보를 관리하고 처리하게 된다.

### 3.2.2 MathML 구조관리기 설계

(그림 4)는 XML 문서 처리부의 XML 관리기 및 XML 노드의 구성을 나타낸다. XML 관리기는 파싱 인터페이스 및 XML 노드를 조작하고 처리할 수 있는 인터페이스를 제공한다. 이벤트 처리기를 통해 처리되는 각각의 XML 노드들은 노드 생성기를 통해 생성되며, 이는 노드 타입에 따라 각각 다른 형태의 노드를 생성하게 된다. 노드의 형태는 문서의 최상위 정보를 나타내는 Document 타입, PI 정보를 나타내는 PI 타입 및 Element, Text, Entity 타입으로 나뉜다. 생성된 노드는 해당 노드에 관련된 정보를 포함하게 되며, 이는 (그림 4)에서처럼 부모 노드 정보, 전후 형제 노드 정보, 처음 자식 노드와 마지막 자식 노드 정보 및 속성 정보를 포함하게 된다. 이러한 정보를 통해 문서의 구조를 순환하여 정보를 검색하고 수정할 수 있게 된다.

(그림 4) XML 파서 모듈의 구성

### 3.3 문서편집 인터페이스 설계

문서 편집부는 MathML의 기본적인 원문 텍스트의 편집 기능을 제공하는 원문 편집 인터페이스와 논리적인 계층구조를 표현하고 이에 대한 편집기능을 제공하는 구조적 편집 인터페이스, 수학적이 디스플레이 되는 부분을 직접 편집하는 기능을 제공하는 디스플레이 편집 인터페이스로 구성된다. 각 인터페이스를 통해 편집되는 데이터는 문서 처

리부로부터 관리되고 정의되는 MathML 노드구조를 통해 처리되며, 각 인터페이스의 특성에 맞는 데이터와 MathML 노드 구조간의 변환을 통해 갱신이 이루어 지고, 각 편집 인터페이스 데이터간의 동기화가 이루어진다.

### 3.4 MathML 디스플레이 편집 인터페이스 설계

MathML 포맷팅 모듈은 크게 토큰 포맷팅 인터페이스와 레이아웃 포맷팅 인터페이스로 나뉜다. 실제 화면의 구성과 배치에 관련된 레이아웃 포맷팅 인터페이스를 통해 MathML의 구조적 정보를 화면상에 반영하고 토큰 데이터가 표현되기 위한 기반영역을 구성한다. 이러한 레이아웃 정보의 하위에는 하나 이상의 토큰정보가 오게 되고 이를 토큰 포맷팅 인터페이스를 통해 처리하게 된다.

(그림 5)는 MathML 포맷팅을 위한 각각의 MFOT들과 관리기를 클래스화 한 모습이다.

(그림 5) MathML 포맷팅을 위한 클래스 구성도

MFOT들은 기본적으로 MFC(Microsoft Foundation Class)와 호환을 위해 CObject 클래스를 상속받아 설계하였다. 또한, XML 파서 모듈에서 구현된 XML노드 클래스를 상속받음으로서, DOM 형태의 노드 구조에 기반한 구조적 접근을 통해 포맷팅 처리를 하도록 하였다. CMathML노드는 레이아웃 포맷팅에 필요한 기본적인 정보를 담고 있는 MFOT의 기반 클래스이다. CMathML노드는 표현 마크업에서 토큰 데이터 부를 제외한 General Layout Schemata와 Script and Limit Schemata, 그리고 Table and Matrix부에서 정의된 수학적 레이아웃 정보를 포맷팅하기 위한 클래스를 상속하며, 이들의 기반이 된다. 이는 (그림 5)에서와 같이 괄호를 표현하기 위한 CMMLFence, 분수를 표현하기 위한 CMMLFrac, 윗첨자를 표현하기 위한 CMMLSuper, 행렬이나 테이블을 표현하기 위한 CMMLTable등으로 구성되어 있다. 그 중에서도 CMMLData는 토큰 데이터를 나타내기 위한 영역을 구성하고 나타내는 클래스로서, 하나의 토큰 정보를 갖는 CMMLToken 클래스 객체를 리스트 형태로 관리하는 CMMLTokenList 객체를 멤버 변수로 갖는다.

MathML 포맷팅을 처리하고 관리하기 위한 CMathML-Manager 클래스는 XML파서모듈에서 파싱처리를 담당하고, XML노드구조를 관리하는 CXMLManager를 상속받는다.

3.4.1 포매팅 객체의 기본 구성

MFOT의 기반 클래스인 CMathML노드는 특정 형태의 레이아웃에 대한 정보에 국한되지 않고, 레이아웃 포매팅에 대한 포괄적이고 기본적인 정보 및 포함 가능한 토큰 정보 등을 갖는다.

토큰 포매팅 객체는 MathML명세서의 표현 마크업에서 정의한 토큰 엘리먼트의 내용을 정의하기 위한 것으로서, 수학적 표현의 가장 최소단위 내용을 처리하며 토큰 그룹정보를 담는 CMMLData 클래스의 하위에 리스트의 형태로 관리된다.

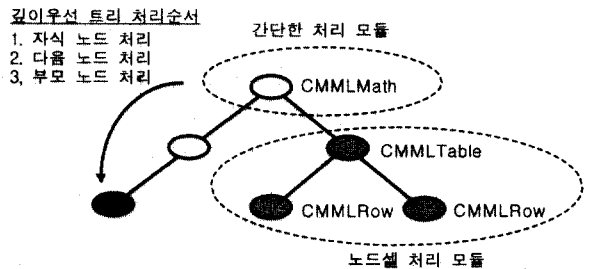
MathML 문서에서는 각각의 토큰 데이터를 구분하고 그 형태와 영역에 따라 세부적으로 엘리먼트를 구성하게 되는데, 이 구조를 그대로 포매팅에 반영하여 처리하기에는 사용자의 입장에서 볼 때 매우 복잡하다. 그렇기 때문에 MathML 노드 구조를 MathML 포매팅 노드로 매핑하는 과정에서 각각의 MathML노드 내의 토큰 데이터를 병합하여, 사용자가 입력할 수 있고 토큰 그룹이 표시되는 영역인 CMMLData클래스의 표현 단위로 처리하였다. 내부적으로CMMLData 클래스 내에서 표현되는 토큰 데이터 목록을 하나의 텍스트 형태로 갖도록 하고, 화면상으로는 CMMLData영역 하나하나의 단위로 구분하였기 때문에 사용자에게 MathML의 문법을 일일이 인식할 필요 없이 쉽게 편집이 가능하도록 하였다. 이러한 내용을 (그림 6)에서 확인해 볼 수 있으며, 이는 2x라는 수학적식을 MathML문법으로 표현하여 XML 노드로 구성하고 다른 인터페이스 데이터로 적용한 모습이다.

은 메모리에 저장될 클래스 객체로 쓰이게 된다. CMathML노드 클래스는 내부적인 모듈로 레이아웃 단위로 디스플레이 하는 모듈과 영역을 그리는 모듈 등을 포함한다.

또한, 포매팅 모듈을 관리하는 클래스인 CMathMLManager는 MFOT를 운영하는 모듈과 앞서 설명한 포매팅을 위한 클래스 객체를 생성하여 메모리 배열로서 관리하는 모듈이다. 또한 페이지를 포매팅 하는 모듈과 전체 포매팅 모듈을 디스플레이하는 모듈을 포함한다.

(1) MFOT 운영

본 모듈은 MathML DOM 노드를 MFOT로 매핑하는 과정에서 생성되어 넘어온 MFOT를 깊이 우선 순위 방법의 탐색 기법을 이용하여 전체 FOT 구조를 파싱하고 포매팅에 필요한 정보로 변환한다. 실행도중 각 노드의 타입에 따라 처리 모듈을 달리 하였다. 처리 모듈은 크게 두 가지 형태로 구분되어 지는데, 하나는 단순 현재 노드의 처리를 위한 모듈이고 또 하나는 현재 노드를 포함하는 하위 노드들의 처리를 위한 모듈이다. (그림 7)은 운영 모듈에 대한 전반적인 처리 과정과 운영 방법을 보여준다. MathML에서 행렬이나 테이블을 나타내는 mtable노드는 하위로 행을 나타내는 하나이상의 mtr엘리먼트가 순차적으로 오듯이, Table형태를 포매팅 하기 위한 CMMLTable클래스 역시 하위로 행을 나타내는 CMMLRow와 열을 나타내는 CMMLCol을 동시에 생성하여 처리해주게 된다.



(그림 7) 운영 과정 및 처리 형태

(그림 6) MathML문법에 따른 XML 노드 구성 및 다른 인터페이스로서의 적용

포매팅 노드로 구조가 변경되는 내용을 통해 알 수 있듯이, MathML의 구조적 문법을 생략하고 사용자가 입력한 CMMLData내의 입력값을 하나의 텍스트 노드로 인식하여 처리하게 된다.

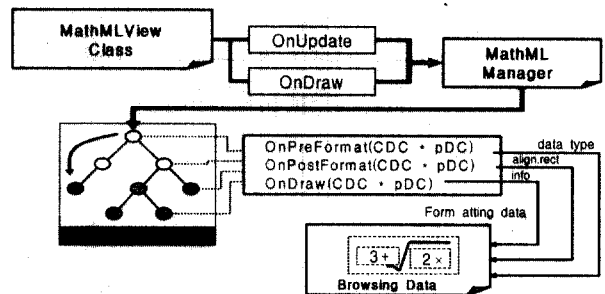
3.4.2 포매팅 처리 알고리즘

본 논문에서 MathML포맷터 처리 모듈은 기본적으로 포매팅을 위한 클래스와 포매팅 모듈을 관리하는 클래스로 구분되어 진다.

포매팅을 위한 클래스는 위에서도 언급했듯이, 크게 토큰 단위와 레이아웃 단위로 나누어지며, 각각 CMathML노드 클래스와 CMathMLToken클래스를 통해 설계되었다. 이들

(2) 레이아웃 포매팅

레이아웃 포매팅 모듈은 토큰 데이터가 표현되기 위한 영역의 설정과 레이아웃 데이터를 포매팅 한다. 전체적인 포매팅의 흐름은 크게 포매팅의 기반을 처리하여 제공해주



(그림 8) 레이아웃 포매팅 흐름도

는 부분과 실제 포맷팅을 담당하는 부분으로 나뉜다. 이는 각각 수학식이 포맷팅되는 뷰영역을 담당하는 CMathML-ProcView 클래스의 OnUpdate메소드와 OnDraw메소드에서 시작되며, (그림 8)과 같은 흐름을 통해 이루어진다.

(3) 토큰 포맷팅

토큰 데이터를 포맷팅하기 위해 설계된 CMathMLToken 클래스는 MFOT객체 중 토큰그룹과 그 포맷팅 영역을 정의하기 위한 CMMLData 클래스 하위에 표현되며, 하나의 토큰을 구성하는 하나 이상의 문자 데이터와 그에 관련된 여러가지 정보를 갖는다. CMathMLToken클래스 객체는 CMMLData클래스에서 리스트의 형태로 관리되며, 포맷팅 기반 처리 모듈인 OnPostFormat메소드를 통해 이 리스트의 검색이 이루어져 순차적으로 토큰 데이터 정보를 추출하고 포맷팅하게 된다. 이 과정에서 문자의 타입에 따라, 각각의 문자의 영역 정보를 설정하고, 속성값에서 정의된 내용에 따라 스타일 정보가 적용된다. 이렇게 설정된 CMMLData클래스 객체 영역내의 토큰 데이터들은 다시 한번 Draw메소드와 OnPostFormat 메소드를 통해 설정된 각 토큰의 영역정보와 폰트 스타일 정보를 적용하여 하나하나의 토큰 정보를 포맷팅 해주게 된다[9, 10]. CMMLData 객체는 사용자에게 해당 영역에 직접 키보드를 통해 토큰 정보를 변경할 수 있도록 하는 편집상자와 유사한 인터페이스를 제공하며, 이와 관련된 디스플레이 편집에 관련된 구현은 뒷부분에서 소개한다.

(4) 디스플레이 편집

디스플레이 편집 모듈은 기본적으로 MathML문법에 유효하게 포맷팅된 MathML 뷰영역을 키보드 입력을 통한 토큰 데이터 변경과 도구모음에 정의되어 있는 특수문자나 레이아웃 정보를 적용하여 편집하는 인터페이스를 제공하고, 그 결과가 곧바로 뷰영역에 표현이 가능하도록 처리하는 모듈이다. 키보드를 통한 편집은 현재 편집 중인 영역을 나타내기 위한 캐럿(Caret) 표기 모듈과 키보드 이벤트를 통한 포맷팅 갱신 모듈로 나뉜다.

(그림 9) 캐럿의 표현과 관련 내용

4. 구현 및 고찰

본 시스템은 IBM-PC 호환 컴퓨터 (Pentium III-800) 에서 개발하였으며, Windows 2000 Server 의 운영체제 환경

에서 Microsoft Visual C++ 6.0 (Developer Studio 98 Enterprise Edition)에 Service pack 5를 사용하여 구현하였다.

(그림 10) 전체 시스템 화면 구성

(그림 10)은 전체 시스템의 화면 구성을 보이고 있다. 좌측의 트리창은 구조적 편집 인터페이스이고, 하단의 에러 보고창은 MathML 문서를 편집시에 MathML의 유효한 문법과 구조에 기준 하여 요소나 속성 등의 오류에 대해 알려주는 역할을 한다. 우측 중간에 있는 창은 기본적인 MathML 형식의 편집 인터페이스를 제공하며, 우측상단의 창은 수학식을 디스플레이 해주고, 이를 직접 편집할 수 있는 디스플레이 편집 인터페이스를 제공한다. 마지막으로, 수학식 및 기호를 입력하기 위한 한 쌍의 도구모음이 상단에 위치해 있다.

4.1 구조적 편집

(그림 11) 구조적 편집의 예

구조적 편집 인터페이스는 (그림 11)과 같이 편집을 원하는 노드에서 마우스 클릭 이벤트를 통해 나타나는 팝업 메뉴를 통해 편집 기능이 제공되며, 이 과정에서 선택된 노드의 상세한 정보는 왼쪽 밑의 프로퍼티 창을 통해 표시된다. 특정 노드에 새로운 엘리먼트나 속성을 추가하는 경우, 해당 노드의 구조적 정보를 기반으로 하는 각기 다른 프로퍼티 편집 기능을 제공하고, 하위 노드를 삽입하는 경우에는 MathML DTD 참조를 통해, 하위로 올 수 있는 요소나 속성의 리스트들만을 팝업 메뉴에 나타내고, 이들 중에 원하는 데이터를 선택하도록 하기 때문에, DTD를 참조하지 않고도 쉽게 유효한 문서를 작성할 수 있도록 해준다.

## 4.2 디스플레이 편집

(그림 12) 디스플레이 편집을 통한 스타일 속성 변경

(그림 12)는 본 시스템을 통한 MathML 문서의 디스플레이 편집과정을 보인다. 그림에서 보는 바와 같이 브라우징된 수학적 특성의 특정 영역을 스타일 속성 편집 다이얼로그를 통해 색상이나 폰트정보의 변경이 가능하다. 디스플레이 창에서 키보드로 직접 수학적식을 입력하여, 이를 구조정보와 원문정보에 반영하기 때문에 MathML 지식을 갖추지 않더라도 MathML 문서의 편집이 가능하다.

(그림 13)은 특수문자입력 도구모음을 통해 수학적식을 입력하는 모습이다. 위쪽의 특수문자입력 도구모음과 수학적킴포넌트 입력 도구모음을 통해 사용자가 키보드를 통해 입력이 불가능한 수학적 요소들을 입력하게 된다.

(그림 13) 특수문자 입력을 통한 디스플레이 편집

(그림 14)는 복잡한 수학적식을 표현하는 MathML과 이를 본 시스템에서 처리한 결과를 보여주고 있으며 수식만을 확대한 그림을 따로 보여주고 있다.

(그림 14) 복잡한 수학적식을 실행한 결과화면, 확대한 화면(우)

(그림 15)에서는 위에서와 같은 수학적식을 Amaya에서 실행한 결과이며 수식만을 확대한 것을 보여주고 있다.

(그림 15) 복잡한 수학적식을 Amaya에서 실행한 결과화면, 확대한 화면(좌)

## 5. 고 찰

본 논문은 MathML 문서의 파싱을 통해 MathML 문서에 정의된 구조적 정보와 데이터 정보를 갖는 MathML 노드 Tree를 생성하고, 이를 편집 인터페이스를 통해 데이터를 생성 및 수정하여 MathML 문서 포맷팅을 위한 MathML Formatting Object Tree로 변환하는 모듈을 설계 및 구현하였다. 이를 이용하여 MathML 문법에 유효한 형태의 수학적 브라우징을 제공하고 다양한 편집 인터페이스를 제공하는 MathML 문서 편집 및 표현 시스템을 개발하였다.

본 논문에서 개발한 MathML 편집 시스템은 MathML 문서의 검증 및 정보 추출을 위해 DOM 형태의 구조와 처리를 지원하는 자체 개발 파서를 사용하여 MathML 구조 트리를 생성하였으며, MathML 포맷팅을 위한 객체 트리를 표현하기 위한 MathML 포맷터를 구현하였다.

MathML 포맷팅 처리를 위한 MathML 포맷터는 다른 시스템에 쉽게 적용이 가능한 형태의 프로토타입 모듈로 정의 및 구성하였고, 포맷팅 모듈 처리과정을 최소화하여 기존의 시스템을 사용함으로써 필요로 하지 않는 모듈까지 처리하는 부담을 덜어줌으로써 속도의 향상을 갖도록 구현하였다. 또한 수학적식을 표현하는 결과의 품질에 대해서는 본 시스템에서 처리한 결과와 기존의 시스템인 Amaya에서 처리한 결과를 (그림 14)와 (그림 15)에서 각각 보여주고 있다. 본 시스템에서 화면의 확대 축소시 자연스러운 결과를 보여주고 있다.

본 시스템의 장점은 첫째, MathML 문서의 검증 및 정보 추출을 위한 기존의 파서를 배제하고, 이를 직접 구현함으로써 외부 모듈로부터의 의존도를 최소화하고, 속도 향상을 가져온다는 점이다. 둘째, 웹 표준화 기구인 W3C에서 제안하는 MathML의 포맷팅 처리에 대한 구성을 기반으로 구현함으로써 표준화에 입각한 처리 시스템으로서 변화에 능동적으로 대처 가능하고, 셋째로는 시스템을 모듈화 하여 부분적인 수정 및 대체가 쉬워 누구나 언제든지 이용 가능하다는 점이다. 단점으로 디스플레이 편집부에서의 복사, 잘라내기, 붙여넣기 등의 편집 기능이 미흡하고 세부적인 사항의 스타일 처

리의 보완이 필요하다. 시스템 개발에 사용한 W3C의 MathML은 2001년 2월 1일자 Recommendation 단계의 제안을 기준으로 구현하였으며, 본 시스템은 W3C의 MathML 명세가 갱신될 경우나 추가 기능 제공을 위해 확장할 수 있도록 설계하였다. 또한, MathML은 아직 계속 개발 중이므로 앞으로 지속적인 관심과 연구가 요구된다. <표 1>에 다른 MathML 편집시스템간의 특징 및 장단점을 비교하였다.

<표 1> 타 MathML 편집시스템간의 비교

시스템	특징 및 장점	단 점
Amaya	<ul style="list-style-type: none"> <li>W3C에서 인증한 MathML을 지원하는 가장 일반화된 브라우저/편집기</li> <li>윈도우즈, 유닉스 등 다양한 운영체제 지원</li> <li>공개소스</li> </ul>	구조적 형태의 편집 인터페이스 미비
Mathematica	<ul style="list-style-type: none"> <li>MathML 편집 및 수학에 관련된 강력하고 다양한 기능을 지원하는 수학적 복합 어플리케이션</li> <li>비주얼하고 편리한 인터페이스</li> </ul>	처리 속도가 느림
MathType	<ul style="list-style-type: none"> <li>디스플레이 편집 인터페이스 지원</li> <li>윈도우즈, 유닉스 등 다양한 운영체제 지원</li> <li>타 어플리케이션의 플러그인 형태로 사용가능</li> </ul>	문서의 소스편집 불가, 브라우저 형태에만 기준을 둬
본 시스템	<ul style="list-style-type: none"> <li>수학식이 브라우저된 화면을 직접 편집가능</li> <li>모듈의 최적화로 빠른 렌더링 속도</li> <li>사용자 중심의 다양한 인터페이스 제공</li> </ul>	복사/붙여넣기 등의 편집기능 미비, 스타일 처리 개선필요

**6. 결 론**

본 논문에서는 올바로 편집된 MathML에 기반한 수학적 편집처리를 웹상에서 수학식이 브라우징 되도록 하였고, 화면에 브라우징된 수학적식을 직접 편집 가능하도록 인터페이스를 제공함으로써 MathML 문서를 보다 쉽고 효율적으로 편집할 수 있도록 사용자 중심의 다양한 인터페이스를 제공하였다. 뿐만 아니라, 내부적으로는 기존 XML파서의 의존도를 줄이고, 직접 설계 및 구현한 파서 부분을 통해 MathML문서를 편집하거나 브라우징 할 때 처리하는 속도에 있어서 향상을 가져왔고, W3C에서 제안한 포매팅 처리를 구현함으로써 변화에 유연하게 대처할 수 있는 장점을 가지고 또한 모듈화되어 있어서 쉽게 수정이 가능하다는 장점을 가지고 있다. 따라서 개발자뿐만 아니라 일반 사용자도 쉽고 효율적으로 웹상에서 수학적식을 편집하고 브라우징 하는 것을 가능하게 하였다.

본 편집 시스템은 웹상에서 수학적식 사용의 보편화에 기여하고, XML을 기반으로 한 수학적식을 포함하는 문서의 효율적인 처리 등에 유용하게 사용될 것이다.

**참 고 문 헌**

[1] W3C, Mathematical Markup Language Ver 2.0, <http://www.w3.org/TR/MathML2>.  
 [2] W3C's Math Home Page, <http://www.w3.org/Math/>.  
 [3] MathML-What's in it for us?, <http://tech.irt.org/articles/>

js081/.  
 [4] Microsoft XML SDK Documentation, <http://msdn.microsoft.com/xml/default.asp>  
 [5] W3C's Document Object Model, <http://www.w3.org/DOM>.  
 [6] IBM Techexplorer, <http://www-4.ibm.com/software/network/techexplorer/>.  
 [7] Amaya - W3C's Editor/Browser, <http://www.w3.org/Amaya/>.  
 [8] W3C, Associating Style Sheets with XML Documents Version 1.0, <http://www.w3.org/TR/XML-stylesheets/>.  
 [9] W3C, eXtensible Style Language(XSL) Specification, <http://www.w3.org/TR/WD-xsl/>.  
 [10] Boumphrey Frank, "Professional XML Applications," WROX Press, 1998.  
 [11] MSDN Online(XML), <http://msdn.microsoft.com/xml/>.  
 [12] William J.Pardi, "XML in Action Web Technology," Microsoft Press, 1999.  
 [13] Alex Homer, "XML IE5 Programmer's Reference," Wrox Press, 1999.

**강 치 원**

e-mail : kangchi0@hanmail.net  
 1993년 광운대학교 전자계산기공학과 졸업 (학사)  
 1995년 광운대학교 대학원 전자계산기공학과 졸업(석사)  
 2000년~현재 배재대학교 컴퓨터공학과 박사과정 재학  
 1995년~현재 한전KDN(주) 재직  
 관심분야 : XML, ebXML, 멀티미디어 문서처리, SGML

**전 성 학**

e-mail : junsh@ocic.co.kr  
 1991년 광운대 컴퓨터공학과 학사  
 1993년 광운대 컴퓨터공학과 석사  
 1996년 광운대 컴퓨터공학과 박사과정 수료  
 1991년 현대미디어시스템 입사  
 1997년 현대정보기술입사  
 2000년 OCI 정보통신 연구소 입사  
 현재 OCI 정보통신 연구소 책임연구원  
 관심분야 : SGML, HiTime, XML, XSL

**정 회 경**

e-mail : hkjung@mail.pcu.ac.kr  
 1985년 광운대학교 컴퓨터공학과 졸업(학사)  
 1987년 광운대학교 컴퓨터공학과 졸업(석사)  
 1993년 광운대학교 컴퓨터공학과 졸업(박사)  
 1990년 영국 ICL 연구소 연구원  
 1998년~1999년 배재대학교 학술정보처부 처장  
 2001년~현재 배재대학교 멀티미디어 지원센터장  
 2002년~현재 배재대학교 정보통신공학부 컴퓨터공학 전공주임 교수  
 1994년~현재 배재대학교 정보통신공학부 부교수  
 관심분야 : 멀티미디어 문서정보처리, SGML/XML, DSSSL/XSL, XML/EDI, ebXML, Web Service