XML 기반의 수식 표현 및 처리 : MathML

조현주 연구원/ 한국데이터베이스 진흥센터 연구개발팀

데이터베이스에서 기초정보로 포함되는 여러 가지 수학 기호와 수식은 일반 문자들과는 다른 독특한 처리방법을 필요로 한다.

워드프로세서에 포함되어 있는 수식편집기가 이런 기능을 처리하는 대표적 예인데, 과학기술분야에서는 이전부터 TeX과 TeX의 매크로 패키지인 LaTeX의 규칙이 많이 이용되고 있다.

이외에도 한글의 수식편집기에도 사용되는 eqn, SGML 계열의 수식 DTD 등 수식 표현을 위한 문법은 여러 가지가 있다.

과학기술분야의 출판물이나 학술지 제공 서비스는 웹상으로 옮겨가는 추세이며, 다양한 애플리케이션간의 데이터 교환 언어로 XML이 부상하고 있다.

문서에 나타나는 복잡한 수식을 표현하기 위해 XML 기반의 수식 마크업 언어인 MathML이 대표적인 인터넷 표준 기구인 W3C(World Wide Web Consortium)의 표준으로 제정되었다. 이와 함께 관련된 연구기관, 출판사, 소프트웨어 개발회사 등의 연구·개발·지원 활동이 활발해지고 있다.

MathML은 수식을 표현하거나 처리하는데 있어 새로운 방법이 아니라고 볼 수도 있다. 그러나, 다양한 애플리케이션간의 데이터 교환, 상이한 문법들간의 호환, 단순히 이미지로만 처리됨으로써 잃게 되는 수식의 의미 처리가 가능하다는 의의가 있다.

1 MathML 개요

MathML(Mathematical Markup Language)은 인터넷 관련 표준화 활동을 주관하는 W3C의 Math Working Group에서 개발하고 있으며, 2000년 3월 현재 W3C Working Draft 버전 2.0이 발표된 상태이다.

MathML 2.0은 MathML 2 DTD, Operator Dictionary, Markup rule, Character & Entities 등으로 구성되어 있다.

■ xml과 차이

MathML은 XML(eXtensible Markup Language) 기반이면서도 XML 유효 문서보다 강력한 규칙을 적용하고 있다. 일반 XML에는 없는, 함수가 취할 수 있는 변수의 갯수와 유형을 규정하고 있다.

따라서, 일반 XML 브라우저가 아닌 MathML 브라우저가 별도로 필요하다.

■ MathML의 응용사례: Illinois D-Lib Testbed

1) 1994년부터 1998년까지 Grainger Engineering Library Information Center와 University of Illinois가 1차로 국립과학재단, 미국방성, 나사로 구성된 디지털도서관프로젝트에서 자금지원을 받아 진행했고, 이후 CNRI(Corporation for National Research Initiatives)의 디지털도서관 테스트베드로서 2001년까지 진행될 예정이다.
2) 여러 출판사가 가지고 있는 6만건 상당의 대규모 저널 전문 디지털 도서관을 구축하는 프로그램으로, 과학기술분야 저널에서 나타나는 다양한 수식을 표현하기 위해 MathML이 채택되었다.
3) 출판사별로 다양한 XML, SGML기반 수식 마크업을 표준적인 presentational MathML 마크업으로 변환한다. 관련 기술로는

표준 기술인 XSLT(eXtensible Style Sheet Transformation), DOM(Document Object Model), javascript 등을 이용한다.

2 MathML의 구조

■ 혼합 마크업(combined markup)

MathML의 가장 큰 특징은 여타 다른 문법에서 일반적으로 사용되는 표시 기반 문법(presentation markup)만이 아닌 수식의 의미를 엘리먼트로 표현한다는(content markup) 점이다.

예를 들어, 수식 "(a+b)2"이 있을 때, 수식을 표현하기 위해 사용된 "괄호", "a", "b"라는 문자, 숫자 "2"를 "첨자"로 표기하는 것은 종전의 표시 문법이 된다.

반면, 이 수식의 의미, 즉 변수 a와 b의 합의 제곱을 태깅하는 것이 의미 정보 문법이 된다.

두 가지를 혼합한 방법을 사용하는 이유는 다음의 3가지이다.

1) 소프트웨어간의 호환 : 의미 정보 표현법과 표시 표현법을 양극단으로 표시할 때 대부분의 소프트웨어는 그 사이 어딘가에 존재한다.

   이 다양한 프로그램을 표현하기 위해서는 두가지를 혼합한 문법(combined markup)이 필요하다.

2) 수식 문법간의 호환: TeX 등 대부분의 수식표현 문법에서는 수식의 표시(presentation), 성격에 관한 형식만을 규정하는 문법을 규정하고 있어서 수식기호를 사용한 의미의 해석에 대해서는 설명이 없다.

3) 수학기호의 의미를 파악하는 동시에 수학기호간의 관계를 해석하는 과정 즉, 의미 마크업은 데이터의 자동처리에 필요하다. 의미 마크업만으로는 수식표현을 어떻게 보여줄 것인지(렌더링)를 조정할 수 없고, 동일 표현이 다른 수학적 의미를 나타내기도 한다.

   표현 마크업만으로는 다른 상황에서 MathML 문서를 재사용하기 어렵기 때문에 혼합 마크업이 필요하다.

■ 수식 표현 트리 (expression tree)

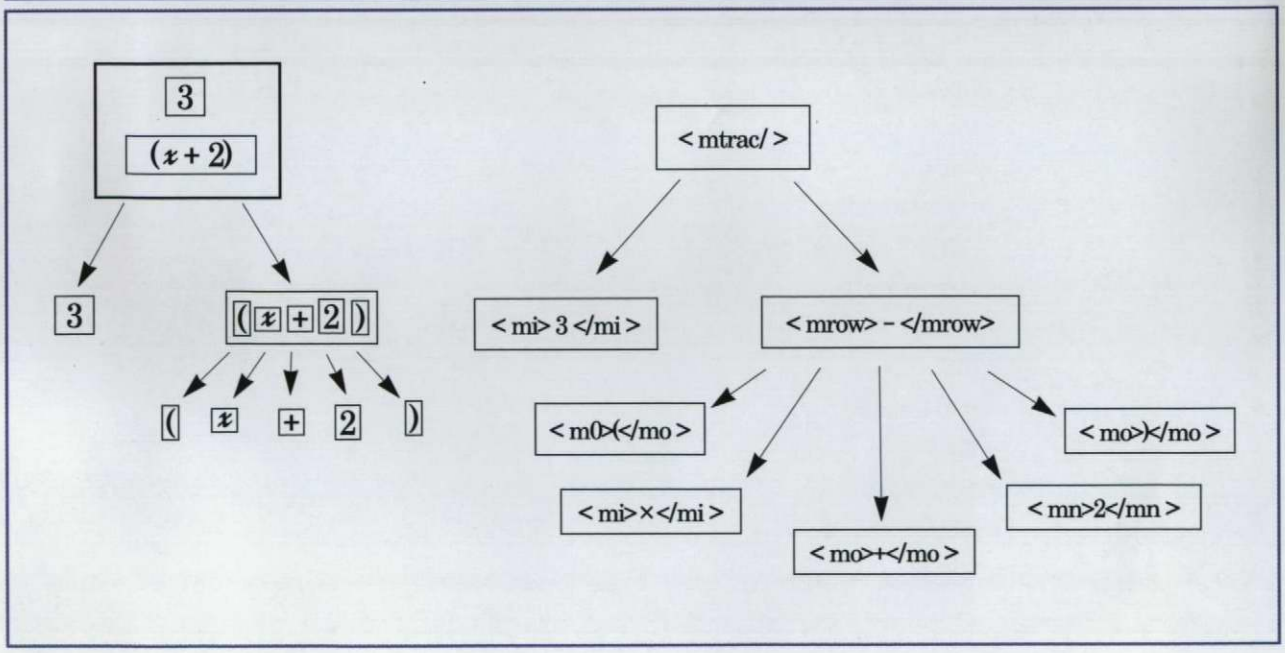
수식 표현은 여러개의 하위 표현으로 나뉘어지며, 최종적으로 최소한의 수식기호로 나뉘어진다.

이 구조를 트리구조로 표현한 것이 수식 표현 트리로서, 데이터를 계층적 구조로 배열한다. 트리구조에서 각 노드는 MathML 엘리먼트와 대응되며, 숫자나 문자 등의 수학 기호나 내용 단위가 된다.

이 트리구조에는 수학 기호와 상징이 반복적으로 사용되는 특징이 반영되어 있다. 수학 기호의 배치는 일정한 순서가 있으므로 트리구조가 적절하다.

이 트리구조에 맞춰 데이터를 배열하고 처리하기 위해 DOM(Document Object Model)을 사용한다. 〈그림1〉

<그림 1> 수식을 Math 트리 구조화한 예(http://www.mathtype.com/support/tutorials/mathml/gitmml/boxes.stm)



< 표1 > 표시 마크업의 기본 엘리먼트

element	description
mi	identifier 식별기호
mn	number 숫자
mo	operation 연산자
mtext	text 텍스트
mspace	space 빈공간
mstring	string 문자열
mglyph	glyph 새로운 기호 첨가

< 표2 > 레이아웃 스키마

class	element	-
script	msub	subscript 아래첨자
	msup	superscript 위첨자
	msubsup	script pair 첨자 세트
	mover	overscript 윗줄 등
	munder	underscript
	munderover	pair
	mmultiscripts	Presubscripts and tensor notation
general	mrow	수식표현의 그룹화
script	mstyle	style
	mfrac	fraction
table	mtable	table, matrix 표나 행렬 표시
	mtr	row 표나 행렬의 열
	mtd	entry in table 표안의 엔트리. mtd 내부에 해당 객체 기술
	mlabeledtr	표의 캡션
maction	maction	수식에 action 부가하기

■ 토큰 엘리먼트의 사용

대부분의 MathML 표현 트리의 노드는 엠프티 엘리먼트나 토큰 엘리먼트이다. 엠프티 엘리먼트는 내용 엘리먼트인 'plus' 등과 같은 수학기호를 표현하며, 유일하게 직접적으로 문자 데이터(character data)를 포함할 수 있는 엘리먼트이다. 문자 데이터는 ASCII 문자와 개체(엔티티: Entity)이다.

수학기호는 ISO 8879 SGML, ISO TR 9573-13을 기본으로 하는 MathML 엔티티 집합에서 정의되어 있다.

<그림 2> 표시마크업의 예

수식 : $x^2 + 4x + 4 = 0$

```
〈mrow〉
  〈mrow〉
    〈msup〉                          ……윗첨자
      〈mi〉x〈/mi〉
      〈mn〉2〈/mn〉
    〈/msup〉
    〈mo〉+〈/mo〉
    〈mrow〉                          ……수식을 최소단위로 쪼갰을 때 단위 수식 표시
      〈mn〉4〈/mn〉
      〈mo〉&InvisibleTimes;〈/mo〉    ……수학 기호 엔티티 삽입
      〈mi〉x〈/mi〉
    〈/mrow〉
    〈mo〉+〈/mo〉
    〈mn〉4〈/mn〉
  〈/mrow〉
  〈mo〉=〈/mo〉
  〈mn〉0〈/mn〉
〈/mrow〉
```

3 MathML 마크업 규칙

■ 표시 마크업 (presentation markup)

표시마크업은 기호법(표시법)의 구조화 (notational structure)라고 볼 수 있다. 표시 기반 마크업은 28개의 엘리먼트와 50개 속성으로 구성되어 있다.

이 대부분의 엘리먼트는 레이아웃 스키마에 대응된다. 각 레이아웃 스키마는 첨자기호체계(super-script or sub-script), 표 같은 2차원 표기(notational device)에 대응된다. (표1, 표2 참조)

1) 엘리먼트

레이아웃 스키마에서 가장 중요한 특징은 하위 스키마 및 엘리먼트의 순서이다.

< 표3 > 의미 정보 마크업의 기본 엘리먼트

element	description
ci	변수명 및 함수명
cn	숫자 (number)
csymbol	(symbol)

엘리먼트의 배열순서는 포맷 변환을 통해 MathML문서를 자동 생성하는데 있어 중요하다.

2) 단점

수식을 트리 구조로 표현할 때 박스형태로 쪼개는 개념을 사용한다. 예를 들어, 분수는 분자, 분모로 나뉘는데, 각각의 행을 〈mrow〉로 태깅한다. 〈mrow〉는 다소 의미가 모호할 수 있으며, 남용되는 경향이 있다.

■ 의미 정보 마크업

표시 마크업을 기호법의 구조화라고 본다면, 의미 정보 마크업은 수식의 의미를 엘리먼트로 표현하기 때문에 수식 구조화 (mathematical structure)라고 구분할 수 있다.

1) 기본 엘리먼트 ; 토큰 엘리먼트를 사용.

기본 엘리먼트는 〈표3〉와 같다.

이 3가지 기본 엘리먼트 외에 75개 엘리먼트와 10개 속성으로 구성되어 있음.

<그림 3> 의미정보 마크업의 예

수식 : $\underset{②}{x^2} + \underset{③}{4x} + \underset{④}{4} \underset{①}{=} \underset{⑤}{0}$

```
〈apply〉
  〈eq/〉                          ……① 함수, 방정식 등 함수의 성격을 먼저 태깅
  〈apply〉
    〈plus/〉
    〈apply〉                      ……②
      〈power/〉
      〈ci〉x〈/ci〉
      〈cn〉2〈/cn〉
    〈/apply〉
    〈apply〉                      ……③
      〈times/〉
      〈cn〉4〈/cn〉
      〈ci〉x〈/ci〉
    〈/apply〉
    〈cn〉4〈/cn〉                   ……④
  〈/apply〉
  〈cn〉0〈/cn〉                     ……⑤
〈/apply〉
```

2) apply : 연산자 표기

엔티티를 삽입해서 간단히 처리하는 방법이 아니라 개개의 엘리먼트로 정의한다. 〈apply〉 태그의 시작 태그와 종료 태그는 연산자 혹은 함수의 범위를 규정한다. 〈apply〉 태그는 반복사용이 가능하며, 혼합연산자를 표현하는데 이용된다.

혼합된 연산자의 경우 각각의 연산의 범위를 구분해주는 기능을 한다. Prefix Notation, 즉 '+', '=' 기호나 연산자를 수식 마크업의 전반부에 표시하는 방법을 취하고 있다. 이 방법에 의하면, 수식을 구성하는 순서를 명확히 하기 위한 괄호를 태그로 나타내줄 필요가 없다.

■ 혼합 마크업

의미정보 마크업과 표시 마크업의 혼합 마크업을 표현하기 위해서는 annotation 엘리먼트를 사용한다. 이외에 MathML 포맷이

<그림 4> 혼합 마크업의 예(presentation makup과 동일한 수식을 content markup으로 표현한 경우)

수식: $\frac{123}{456}$

```
〈semantics〉                                          …… 의미정보 마크업을 표시하는 기본 엘리먼트
  〈apply〉                                            ……수식의 범위를 지시하는 엘리먼트
  〈divide/〉                                          ……나누기
    〈cn〉123〈/cn〉                                    ……숫자
    〈cn〉456〈/cn〉
  〈/apply〉
  〈annotation encoding="Mathematica"〉                ……Mathematica 프로그램에 맞는 코딩
    N[123/456, 39]
  〈/annotation〉
  〈annotation encoding="TeX"〉                        …… TeX 문법으로 수식의 값을 표현
    $0.269736842105263157894736842105263157894\ldots$
  〈/annotation〉
  〈annotation-xml encoding="MathML-Presentation"〉    ……의미정보마크업에 해당되는 표시마크업
    〈mfrac linethickness="2"〉
      〈mn〉123〈/mn〉
      〈mn〉456〈/mn〉
    〈/mfrac〉
  〈/annotation-xml〉
〈/semantics〉
```

아닌 데이터를 표현하기 위해, MathML의 표시 마크업이 아닌 다른 포맷을 사용할 때도 annotation 엘리먼트를 사용한다.

본 자료에서는 동일한 수식을 표시 마크업과 의미 정보 마크업으로 각각 제시하고 〈그림 2, 그림 3〉 의미정보마크업 안에서 표시마크업 주기를 부여한 예〈그림 4〉를 들었다.

4 MathML 데이터 처리 및 활용

■ browser/editor (MathML on Amaya)

표준화 활동은 특정 목적을 위해 스펙을 개발한 후 실제로 구현하고 실험하는 활동이 포함된다. W3C에서도 이를 제대로 구현하기 위해 에디터(겸 브라우저)를 개발한 것이 Amaya이다.

Amaya는 최신 버전인 MathML version 2.0 (2000. 3)을 구현하였으며, 진행중인 STIX 프로젝트의 폰트를 지원하고 있다.

Amaya에서는 수식을 구조화된 컴포넌트로 처리하며, HTML 엘리먼트와 동일하게 처리한다. Amaya는 XML 브라우저라기보다는 Xhtml 브라우저이다. (현재 version.3.2.1) xml 문서를 html 문서로 변환해서 보여주는 동시에 html 문서에 MathML presentation 수식 표현문을 embedding 하는 형식이다.

Equation은 XSL이 아닌 CSS로 처리한다. 또한, Amaya는 정작 중요한 content markup을 구현하지 못하고 presentation markup 문법만을 구현하고 있다.

이 외에도 IBM Techexplorer, EZMath, WebEQ, MS IE, Mozilla 등 다양한 수식 편집 및 브라우징을 위한 소프트웨어에서 MathML을 지원하고 있다.

■ API for MathML

MathML도 xml 문서의 구조 처리를 위해 기본적으로 DOM API를 사용한다. MathML2.0 권고안에서는 core DOM의 확장형 MathML을 제시하고 있다. MathML 데이터를 처리하고 렌더링하기 위해 사용되는 방법은 아직 표준화되어 있지 않은 상태이며, 우선 권고하는 한가지 방법이 DOM 이다.

DOM은 대량의 데이터를 처리하기에는 처리속도가 걸리지만, 수식의 트리구조를 지향하는 MathML의 특성과 매치된다. 문서의 스타일, 폰트 등 외양정보를 표현해낼 수 있다.

현재는 W3C에서도 DOM의 Level 1.0 수준의 지원까지밖에 고려하고 있지 않지만 DOM은 계속 발전하고 있으므로 〈표 4 참조〉 DOM을 사용한 MathML 데이터의 처리도 논리구조 처리, 콘텐트 모델을 제대로 해낼 수 있게 될 것이다. 〈표 5〉는 확장된 DOM의 기본 엘리먼트이다.

< 표4 > DOM의 발전단계

Level 1	Core DOM (XML)
	HTML
Level 2	Style DOM
Level3 (2000. 11. WD 1.0)	Content Model Save & Load DOM

< 표5 > MathML DOM Extension Element

엘리먼트	기능
MathML Element	- MathML 엘리먼트에서 만들어질 수 있는 모든 연산자와 쿼리 정의 - 모든 MathML 엘리먼트에 적용되는 속성의 검색 및 수정 method 제공
MathMLFractionElement	분모, 분자 속성에 접근

5 맺음말

MathML은 다른 분야에서도 데이터 교환의 표준으로 자리매김하고 있는 XML을 기반으로 하고 있다.

MathML은 기계와 기계사이의 커뮤니케이션을 위한 포맷이라는 성격이 강하기 때문에 인간에게 친숙한 인터페이스는 결코 아니다. 현재 기존의 유명한 수식편집기인 EZMath, MathType 등에서 MathML을 지원하고 있으며 W3C, IBM, MS 등 대표적 표준기구와 소프트웨어개발 업체에서 MathML을 지원하는 브라우저, 에디터 등이 개발되어 있으나, MathML을 완전히 구현한 것이 아니다. 또한, Rendering performance 문제도 보완해야 한다. XSLT, DOM 등 XML 데이터처리를 위한 애플리케이션도 계속 개발 중이기 때문에 전반적으로 MathML은 완성된 상태가 아니다.

MathML이 기존의 과학기술분야에서 수식 표현의 우위를 차지하던 TeX나 수식편집기 등을 대체하리라는 전망은 성급한 감이 있다. 기존의 애플리케이션간의 데이터 의미 상실없이 완전한 교환을 가능케할 수 있을 것이라는 당초의 목표대로 과연 구현될지 그리고 어느 정도로 이용될지가 관건이라고 할 것이다.