

사용자 중심의 인터페이스를 지향하는 MathML 편집 시스템

장홍매*⁰ 오수영** 정문영** 조현성** 조현규** 정희경*

*배재대학교 컴퓨터공학과

**한국전자통신연구원 전자상거래 연구부

*(zanghongmei, hkjung)@markup.paichai.ac.kr

**{runtou,mchung,hsc}@etri.re.kr

MathML Editing System with the Intention of User Central Interface

⁰Hong-Mei Zang, Su-Young Oh, Mun-Young Jung, Hyeon-Sung Cho, Hyeon-Kyu Cho, Hoe-Kyoung Jung

*Dept. of Computer Engineering, Paichai University

**ETRI EC Team

요 약

기존 웹상에서의 수학적 처리 및 표기에 관련된 문제점을 보완하고, 수학적 효율적인 표기 및 저장, 공유를 가능하게 하기 위해 W3C(World Wide Web Consortium)에서는 XML(eXtensible Markup Language) 애플리케이션인 MathML(Mathematical Markup Language)을 제정하였다. 본 논문에서는 MathML 문서를 보다 쉽고 효율적으로 편집할 수 있도록 사용자 중심의 다양한 인터페이스 제공을 통해, MathML 지식이 없는 사용자라도 쉽게 편집이 가능하도록 하였으며, 내부적으로는 처리속도 향상을 위해 기존 XML 파서의 기능 다수를 설계 및 구현하였다. 본 편집 시스템은 웹상에서의 MathML 사용의 보편화에 기여하고, XML을 기반으로 하고 수학적식을 포함하는 문서의 효율적인 처리 등에 유용하게 사용될 것이다.

1. 서 론

MathML은 수학적식의 표기형식을 포함하고 수학적식의 구조와 콘텐츠 정보를 정의한 XML 애플리케이션으로, 웹상에서 수학적 정보를 교환 하고 표기 및 처리하는데 목적을 두고 있다. 일반적으로 수학적식을 웹상에서 표현하는 것은, 이미지와 텍스트로 표현되어진다. 그러나, 이는 읽고 표현하기 어려운 뿐만 아니라, 이미지 특성상 상대적으로 큰 데이터 크기로 인해 처리와 교환에 부담을 주게 되고, 수학적식의 검색이나 인덱싱, 그리고 다른 수학관련 어플리케이션에서의 재사용을 어렵게 한다[1][2].

MathML은 이러한 기존의 단점을 극복하고자 수학적 표현과 내용을 마크업 기반의 코드로 정의하였으며, 웹상에서의 수학적 표현과 처리에 효과적인 대안을 제시하고 있다. MathML을 사용하기 위해서는 MathML 문서의 작성과정이 필수적이며, 또한, 올바른 MathML을 위해서는 MathML의 구조 및 복잡한 MathML DTD(Document Type Definition)에 관련된 지식을 요구한다.

이에, 본 논문에서는 MathML 문서에 대해 익숙지 못한 사람들도 쉽게 편집이 가능하도록, 기본 편집기능 뿐만 아니라 MathML 문서를 통해 브라우징 되는 수학적식을 직접 편집할 수 있는 사용자 중심의 인터페이스를 제공하는 MathML 편집 시스템을 설계 및 구현 하였다. 또한, 편집된 MathML 문서를 적용한 수학적식을 브라우징 할 수 있어, MathML 문서의 검증과정을 사용자에게 쉽게 처리할 수 있게 하였다. 뿐만 아니라, 내부적으로는 처리속도 향상을 위해 문서 처리를 위한 파서 부분을 설계 및 구현하여, 기존 XML파서의 의존도를 최소화 하였다.

2. 관련연구

MathML은 표현과 내용, 어느 것에 중심을 두느냐에 따라, 표현(Presentation) 마크업과 내용(Content) 마크업으로 구분된다. 표 1은 기능에 따른 표현 마크업의 분류를 나타낸다.

표 1 기능에 따른 표현 마크업의 분류

Token Element	수학적 표기의 가장 최소 단위를 처리	숫자,문자열, 기호,변수
General Layout Schemata	스타일처리,애러핸들링, 일반영역의 수학적 처리	괄호,분수, $\sqrt{\quad}$...
Script & Limit Schemata	스크립트와 제한된 영역의 처리	$\int, \sum, \prod, \lim \dots$
Table & Matrix	행렬 및 테이블 처리	행렬,테이블

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

3. 시스템 설계

그림 1은 본 편집 시스템의 전체 구성도이다. 입력되거나 초기화된 MathML문서는 문서 처리부의 파서를 통해 파싱과정을 거쳐 DOM(Document Object Model)과 유사한 MathML의 노드구조로 구성되며, 구조 관리기를 통해 관리된다. MathML 노드객체는 구조 관리기를 통해 디스플레이 처리부에 수학적 브라우징에 필요한 정보를 전달하여, 뷰 영역에 브라우징 해주면, 사용자는 이러한 정보와 문서 편집부에서 제공하는 편집 인터페이스를 통해 원하는 형태로

의 MathML 문서 작성이 가능하게 된다.

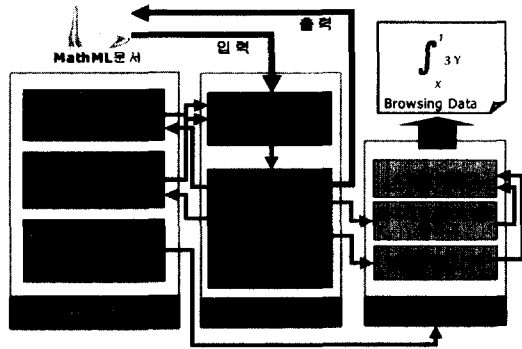


그림 1 시스템 구성도

3.1 문서 처리부

문서 처리부는 MathML 문서의 입출력을 담당하고, 문서 편집부를 통해 갱신되는 각 편집 인터페이스의 데이터를 본 편집 시스템에서 정의한 MathML 노드구조로 재구성 하고, 이를 디스플레이 처리부에 반영하는 역할을 담당한다. 그림 2는 문서 처리부의 구성 및 처리부들을 통해 관리되는 MathML 노드의 구조를 보인다.

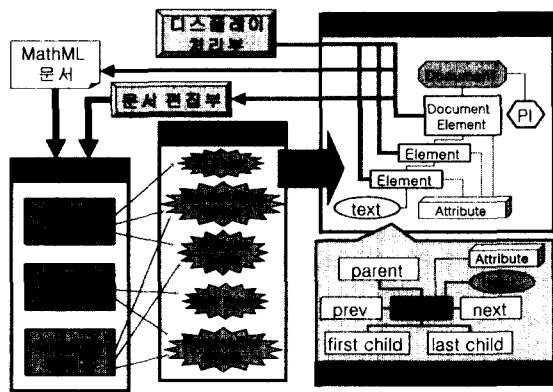


그림 2 문서 처리부의 구성 및 처리노드의 구조

파일의 입력이나 사용자 입력 인터페이스를 통해 갱신된 문자열 형태의 MathML 문서는, 태그 인식기를 통해 문서 내 각각의 태그들을 순차적으로 검색하여 시작태그, 종료태그, 빈태그인지 인식하고 그림 2에서처럼 이벤트 처리기의 이벤트 처리 모듈로 엘리먼트 이름정보, 속성 그룹정보, 텍스트 노드 정보등을 전달하여 SAX(Simple API for XML)와 같은 Event-Driven 방식을 통해 파싱 단계를 거친다. 이 처리 과정에서 인식된 Process Instruction 노드, Element 노드, Text노드 등은 트리 형태와 같은 MathML 문서구조로 구성되고, 구조 관리를 통해 관리된다. 노드 구성 과정에서 MathML 노드가 생성되어 현재 위치의 부모노드 하위에 추가됨과 동시에, 전·후 형제노드 정보, 부모노드 정보 및 첫번째 자식노드와 마지막 자식노드, 그리고 속성정보와 Text노드 정보를 갖게 되고, 이는 구조 관리를

통해 문서 조작에 대한 정보를 제공하게 된다.

MathML문서가 저장되거나, 편집을 통해 문서내용이 수학적 브라우저에 적용되는 경우, 그리고 다른 편집 인터페이스에 갱신된 데이터가 적용되는 경우에 구조 관리를 통해 MathML 구조정보의 재귀적 순환이 이루어 지고, 순환되면서 연계되는 각 노드들의 정보들을 각각의 처리기나 문서에 적용하게 된다.

3.2 문서 편집부

문서 편집부는 MathML의 기본적인 원문 텍스트의 편집기능을 제공하는 원문 편집 인터페이스와 논리적인 계층구조를 표현하고 이에 대한 편집기능을 제공하는 구조적 편집 인터페이스, 수학적식이 디스플레이 되는 부분을 직접 편집하는 기능을 제공하는 디스플레이 편집 인터페이스로 구성된다. 각 인터페이스를 통해 편집되는 데이터는 문서 처리부로부터 관리되고 정의되는 MathML 노드구조를 통해 처리되며, 각 인터페이스의 특성에 맞는 데이터와 MathML 노드구조간의 변환을 통해 갱신이 이루어 지고, 각 편집 인터페이스 데이터간의 동기화가 이루어진다.

3.3 디스플레이 처리부

디스플레이 처리부는 MathML 문서를 입력하거나, 편집을 통해 갱신된 MathML 노드구조의 변경에 맞추어 수학적식을 뷰 영역에 브라우징 해주는 역할을 한다. 그림 3은 디스플레이 처리과정과 디스플레이 처리부를 구성하는 디스플레이 정보 객체의 구성을 보인다.

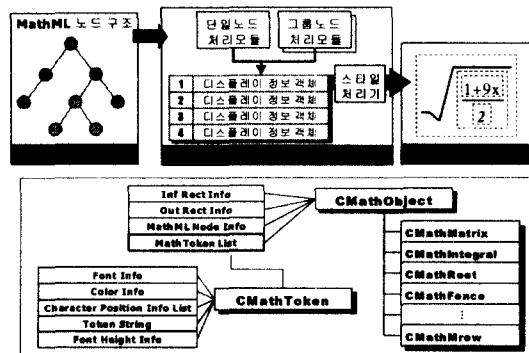


그림 3 디스플레이 처리 과정 및 디스플레이 정보 객체 클래스 구성도

MathML 노드구조의 순환을 통해 얻어온 노드 정보를 통해, 각 노드가 토큰형태의 단일노드로 처리되는 부분인지, 분수나 루트 등과 같이 그룹단위의 노드로 처리되는 부분인지 구분하여, 각각의 수학적 컴퍼넌트나 토큰 데이터를 처리하는 디스플레이 정보 객체를 통해 수학적식을 브라우징 하게 되고, 이 과정에서 스타일 처리기를 통해 MathML문서에서 정의한 스타일 속성이 적용된다. 디스플레이 정보 객체는 그림 3에서처럼 사각형 점선 안의 하나의 영역을 기본적인 객체 정보로 갖게 되며, 하위로 다른 영역정보를 포함하는 것이 가능하다. 이는 CMathObject 라는 이름의 기본 클래스에 정의되며, 이 클래스에는 해당 노드 및 하위

영역을 포함하는 전체적인 영역정보, 해당 노드만의 영역정보, MathML노드 정보와 같이 수학적 브라우저를 위해 기본적으로 필요한 정보를 멤버 변수로 갖는다. 뿐만 아니라, 디스플레이 영역 정보는 여러 개의 토큰을 통해 구성되는데, 이는 CMathToken 클래스를 통해 정의되며, 리스트의 형태로 CMathObject 클래스의 멤버변수로 정의된다. CMathToken 클래스는 토큰 브라우저를 위한 기본적인 정보인 폰트 및 컬러정보, 토큰을 구성하는 문자의 위치정보 등을 멤버 변수로 갖는다. 수학적 식의 형태에 따라 각기 다른 포함관계나 영역정보를 갖게 되는데, 그림 3과 같이 CMathObject 클래스를 상속받는 여러 확장 클래스를 통해 구현된다.

4. 구현

본 논문에서 구현한 MathML 편집 시스템은 수학식의 브라우징 및 디스플레이 편집 인터페이스를 제공하는 수학적 디스플레이 창, MathML문서구조를 트리형태로 나타내고 구조적 편집 인터페이스를 제공하는 문서 구조 창, 원문 편집 인터페이스를 제공하는 원문 편집 창, 그리고 문서 구조 창을 통해 나타나는 노드의 정보들을 보여주는 프로퍼티 창, 문서 편집시 발생하는 오류나 상황을 표시해주는 에러 보고 창으로 구성되며 세부기능은 다음과 같다.

4.1 디스플레이 편집

그림 4는 본 시스템을 통한 MathML문서의 디스플레이 편집과정이다. 그림에서 보는 바와 같이 브라우징된 수학적 식의 특정 영역을 스타일 속성 편집 다이얼로그를 통해 색상이나 폰트정보의 변경이 가능하다. 디스플레이 창에서 키보드로 직접 수학적 식을 입력하고, 위쪽의 특수문자입력 툴바와 수학적 컴퍼넌트입력 툴바를 통해 키보드로 입력할 수 없는 수학적 요소들을 입력하여, 이를 구조정보와 원문정보에 반영하기 때문에 MathML지식을 갖추지 않더라도 MathML 문서의 편집이 가능하다.

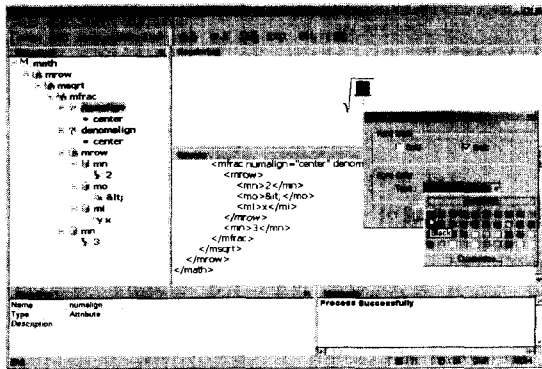


그림 4 디스플레이 편집의 예

4.2 구조적 편집

구조적 편집 인터페이스는 그림 5와 같이 편집을 원하는 노드에서 마우스 클릭 이벤트를 통해 나타나는 팝업메뉴를

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, 하위노드를 삽입하는 경우에는 그림 5에서 처럼 MathML DTD를 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.

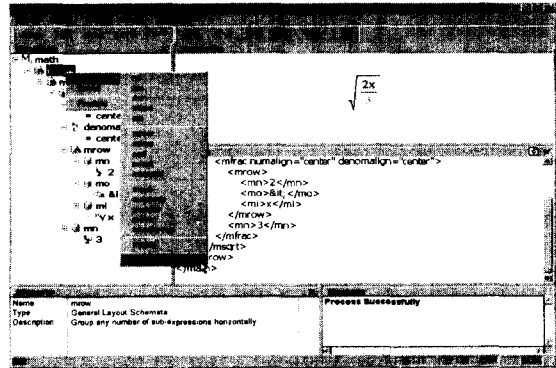


그림 5 구조적 편집의 예

5. 결론

본 논문에서는 올바로 편집된 MathML 문서는 곧바로 적용되어 수학적 식이 브라우징 되도록 하였고, 수학적 식의 브라우징 화면을 직접 편집 가능한 인터페이스의 제공을 통해 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 편집 인터페이스와 수학적 식의 브라우징에 적용하였기 때문에, 사용자 뿐만 아니라 개발자에게도 효율적이고 MathML처리를 가능하게 하였다.

본 편집 시스템은 웹상에서의 MathML 사용의 보편화에 기여하고, XML을 기반으로 하고 수학적 식을 포함하는 문서의 효율적인 처리 등에 유용하게 사용될 것이다.

참고 문헌

- [1] W3C's Math Home Page
<http://www.w3.org/Math/>
- [2] MathML - What's in it for us?
<http://tech.irt.org/articles/js081/>
- [3] W3C's Document Object Model
<http://www.w3.org/DOM>
- [4] IBM Techexplorer
<http://www-4.ibm.com/software/network/techexplorer/>
- [5] Amaya - W3C's Editor/Browser
<http://www.w3.org/Amaya/>