

# 수식을 포함한 전자문헌의 구조적 처리를 위한 XML 문서편집시스템

XML Document Editing System for Structural Processing  
of the Digital Document to Including Mathematical Formula

윤화목(Hwa-Mook Yoon)\*, 유범종(Beom-Jong Yoo)\*\*  
김창수(Chang-su Kim)\*\*\*, 정희경(Hoe-Kyung Jung)\*\*\*\*

## 초 록

현재 기관이나 조직 내에 수많은 양의 데이터가 축적되어 존재하고 있으나 대부분의 데이터는 각 기관이나 조직에 따라 정형화된 형태로 남아있는 실정이다. 정형화된 정보는 정보의 교환 및 공유에 어려움이 있다. 이러한 단점을 극복하고자 지식정보자원관리라는 새로운 개념이 도입되었으며, 축적된 데이터들을 공유 및 관리하기 위한 지식정보자원의 디지털화가 실행되고 있다.

특히 과학기술 또는 교육학술 분야에서는 지식정보자원의 교환 및 공유에 필요한 데이터를 구조적으로 처리하고자 XML을 도입하려는 움직임이 일고 있으며, 이들 분야의 전자문서 안에 사용되어지는 수많은 수학적 표현이 이미지나 텍스트 등의 비구조적인 데이터로 처리됨에 따라 검색과 인덱싱 또는 재사용성 등의 제한사항이 발생하게 된다. 이를 극복하고자 MathML을 이용한 수학적 표현의 처리에 관심이 집중되고, MathML을 구조적인 문서상에 쉽고 효율적으로 처리할 수 있는 솔루션이 요구되고 있는 실정이다.

이에 본 논문에서는 지식정보자원을 목적으로 하는 전자문서의 구조적인 처리를 용이하게 하고, MathML에 대한 전문적인 지식이 없어도 구조적인 문서상에 쉽게 MathML을 생성 및 표현할 수 있는 XML 문서 편집 시스템을 구현하였다.

## ABSTRACT

A lot of accumulated data of many quantity exist within a institution or a organization, but most data is remained in form of standardization as each institution or organization. There are difficulty in exchange and share of information. New concept of knowledge information resource management to overcome this disadvantage was introduced, and the digitization of knowledge information resources to share and manage

\* 한국과학기술정보연구원(hmyoon@kisti.re.kr)

\*\* 한국과학기술정보연구원(ybj@kisti.re.kr)

\*\*\* 배재대학교 컴퓨터공학과(ddoja69@mail.pcu.ac.kr)

\*\*\*\* 배재대학교 컴퓨터공학과(hkjung@mail.pcu.ac.kr)

■ 논문 접수일 : 2002. 11. 13

■ 게재 확정일 : 2002. 12. 11

accumulated data is been doing.

Specially, in science technic or education scholarship it, the tendency that importing XML to process necessary data to exchange and share of knowledge information resources structurally, and limitation of back for search and indexing or reusability is happened according as expression of great many mathematics used inside electron document of these sphere is processed to nonstructural data of image or text and so on. There is interest converged in processing of mathematics that use MathML to overcome this, and we require the solution to be able to process MathML easily and efficiently on structural document.

In this paper, designed and implemented of XML document editing system which easy structural process of electronic document for knowledge information resources, and create and express MathML easily on structural document without expert knowledge about MathML.

키워드: MathML, XML, 지식정보자원관리, 수식편집기, XML편집기, math, mathematical expression

## 1 서 론

정부에서는 지난 99년부터 과학기술, 교육학술, 문화, 역사 등 전략분야를 중심으로 보존이나 이용가치가 있는 국가지식정보자원에 대한 지속적인 디지털화 사업을 추진해왔고, 그 결과로 국가지식정보자원관리센터와 분야별 종합정보센터 등을 지정하여 지식정보자원관리체제를 갖추게 되었다(한국전산원 2001). 하지만, 현재 기업이나 기관 등에 산재해 있는 수많은 지식정보를 교환하기 위한 기본 데이터는 정형화된 형태로 되어 있기 때문에 데이터 확장에 제한이 따르고, 구조적인 데이터 검색 및 처리가 난해한 실정이다. 따라서, 지식들이 생성, 공유, 관리, 보관되고 이를 보다 체계적이고 정교하게 구축하여 이용자가 보다 쉽게 사용할 수 있는 정보기술이 요구된다. 특히, 과학기술이나 교육학술 분야에 있어서 문서상에 수학적 표현은 필수적이지만 이러한 수학적 처리가 현재로써는 이미지나 텍스트 등의 비구조적인 형태로 처리됨에 따라 사용자에게 따른 가독성에 제한이 있고 수학적 검색, 교환 및 처리가 난해한 실정이다.

이에 W3C(World Wide Web Consortium)에서는 문서와 데이터의 교환이 쉽고, 문서를 다양하게 구조화시킬 수 있는 XML(eXtensible Markup Language)을 제정하였다(W3C 1998 <http://www.w3.org>

/TR/REC-xml). XML은 사용이 쉽고, 웹 환경을 지원하며, 하드웨어 및 소프트웨어의 종류에 관계없이 이기종간에서도 장치에 독립적으로 문서를 교환할 수 있다. XML은 여러 어플리케이션들의 표준화된 정보 전달의 매개가 되는 표준으로 지식관리의 구조적 체계를 공유할 수 있는 기반을 만들어 준다. 또한 전자문서상에서 수학적 표현을 효과적으로 처리할 수 있는 대안으로 MathML(Mathematical Markup Language)을 제정하였으며 지식정보자원관리를 목적으로 하는 해당 분야의 수학적 처리에 있어 표준으로 자리잡아 가고 있다(W3C 2001. <http://www.w3.org/TR/MathML2>).

그러므로, 과학기술 또는 교육학술 분야 등에서 XML과 MathML을 정보자원 관리에 도입하고자 하는 움직임이 일고 있고, 이들을 처리할 수 있는 문서편집 솔루션이 요구되고 있다(Richard Anderson 외 2000, 정회경, 홍성찬, 이수연 2001).

이에 본 논문에서는 수학적 표현을 요구하는 정보자원관리의 각 분야에서 사용되는 데이터나 전자 문헌들을 구조적으로 처리하기 위한 XML 편집 시스템을 설계 및 구현하여 각 조직별로 데이터 및 전자문헌을 교환함에 있어 상호운용성을 증대시키기 위한 기반을 제공한다. 또한 저작된 XML 문서 내에 포함될 MathML 수식구조를 전문적인 지식의 필요 없이 사용자 표현에 따라 수식구조를 생성하고 삽입해주는 MathML 수식편집 시스템과 웹을

통하여 수학적 표현하기 위한 방법을 제안한다. 논문의 구성은 다음과 같다. 2장에서는 본 시스템의 이해를 위한 지식정보자원관리에 대한 기본 개념과 XML의 중요성에 대해 알아보고, 3장에서는 본 시스템의 설계 및 처리에 관한 주요 처리 모듈에 관해 알아본 다음, 4장에서 본 시스템에서 실제로 설계한 부분이 어떤 형식으로 보여지고 처리되는지에 대한 구현을 살펴보고, 마지막으로 5장에서 고찰 및 결론을 맺는다.

## 2 관련 연구

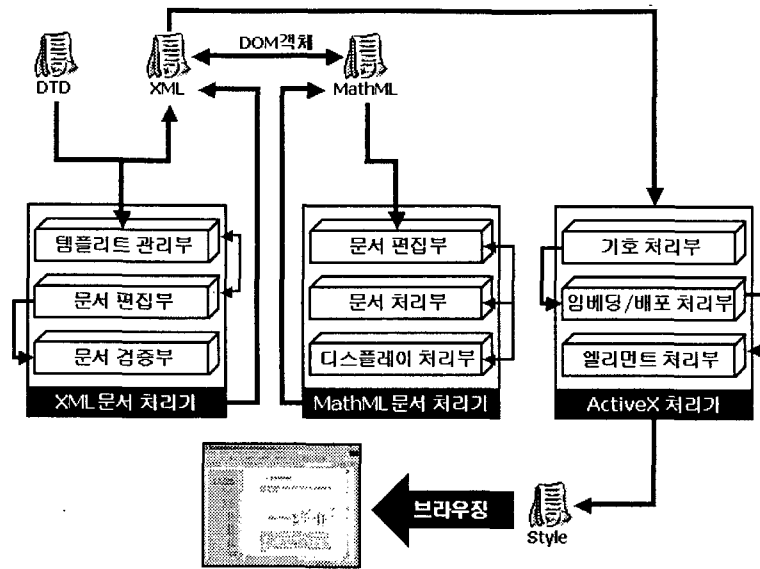
### 2.1 지식정보자원관리의 개념

지식정보자원의 관리란 지식정보의 창출, 축적, 유통, 공유/활용 등으로 구성되는 지식정보활동이 효율적으로 이루어지도록 체계화하고 관리하는 모든 활동을 말한다. 즉 국가의 비전이나 목표의 달성을 위하여 보존 및 내용적, 이용적 가치가 있는 공사부문의 지식정보를 생성 및 선정한 다음, 다양한 정보기술 및 네트워크 인프라를 활용하여 국가 구성원이 용이하게 활용, 재창출할 수 있도록 도와주는 제반활동을 뜻한다. 따라서 국가 지식정보자원관리의 목표는 정부와 기업을 포함한 사회전반의 생산성 증진을 통하여 국가 경쟁력을 제고 하고 일반 국민들의 지식정보자원에 대한 접근성 및 활용성을 혁신하는데 있다.

지식정보자원 관리의 전략적 구성요소는 지식정보자원관리의 대상이 되는 지식정보와 효율적인 지식정보자원관리를 지원하는 H/W, S/W, 유통망 등을 포함하는 정보기술 인프라, 그리고 지식정보자원관리를 지원하는 기반환경 등으로 구성된다.

### 2.2 XML의 사용에 따른 이점

웹상에서 구조화된 문서의 교환 및 응용프로그램에서 보다 쉽게 처리하기 위해 W3C에서 1998년에 제정된 XML 1.0이 웹 문서의 표준으로 대두되면서 각 시스템에 상이한 포맷으로 존재하는 문서들을 통합, 관리하는데 유용하게 사용하게 되었다. 웹을 통해 e-비즈니스가 이루어지고 있는 상황에서 HTML, HWP, MS 워드 등의 데이터 구조에 대한 한계성을 가진 정보로는 수많은 시스템들로 연결된 플랫폼들에서 실제로 정보의 교환이 이루어지기가 힘들다. XML은 조직이나 기관의 데이터 통합을 위한 개방형 표준을 사용하는 기반으로 어플리케이션간 또는 데이터간의 통합을 통하여 일관성을 가질 수 있는 프레임워크를 제공한다. 또한 플랫폼에 독립적이기 때문에 비구조적이거나 구조적인 문서들에 대해 시스템을 변경하지 않고 DTD(Document Type Definition)의 변경만으로 확장하여 사용할 수 있다. 즉, 데이터가 시스템간에 공유될 때 XML은 기존 데이터 구조를 바꾸지 않고서도 새로운 확장을 이룰 수 있기 때문에 데이터베이스 등의 재설계가



<그림 1> 전체 시스템 구성도

요구되지 않으며 어플리케이션 역시 다시 작성될 필요가 없으므로 데이터의 상호운용성 측면에서 상당한 이점을 발견할 수 있다.

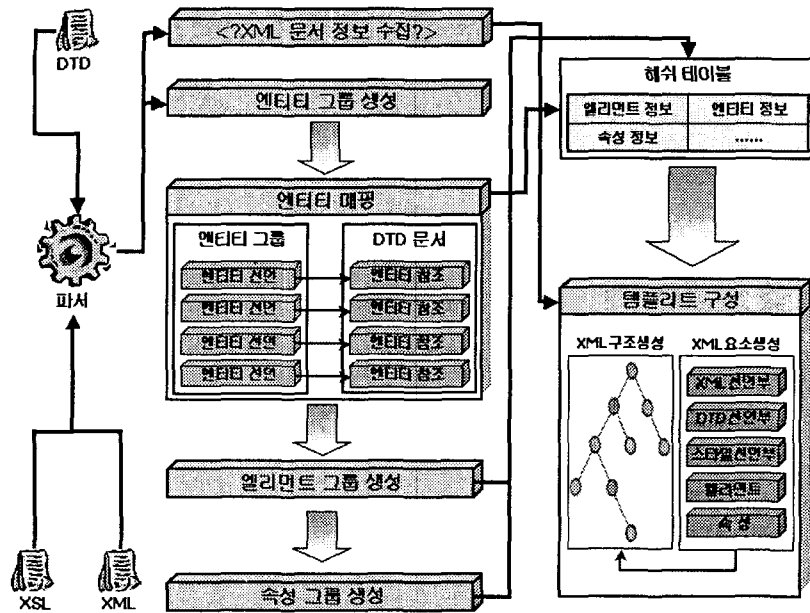
또한, XML은 의미가 있는 정보를 담을 수 있다. XML은 HTML과 같이 표현 중심이 아닌 데이터의 의미나 구조를 다룰 수 있기 때문에 데이터를 보다 유연하게 사용할 수 있다. 이를 통해 사용자가 원하는 일부 정보만을 따로 조합하여 새로운 의미의 문서를 생성하고 활용할 수 있어 뛰어난 재사용성을 가진다. 지식정보자원관리에서의 중요한 것 중에 하나는 관리되고 있는 지식을 활용할 수 있도록 적절한 검색이 가능해야 한다는 것이다. 이에 XML은 현재 제공되고 있는 대부분

의 검색 엔진들에 비해 좀더 세분화된 정보 검색을 가능하게 한다.

### 3 시스템 설계

본 시스템은 XML 기반의 지식정보자원관리시스템에서 요구되는 메타데이터나 XML 문서를 쉽고 효율적으로 구조화하기 위한 시스템으로 크게 XML 문서 처리기와 MathML 문서 처리기 그리고 ActiveX 처리기로 분류한다.

<그림 1>은 본 시스템의 전체 구성을 보인다. XML 문서 구조를 정의한 DTD나 저작된 XML 문서를 XML 문서 처리기에 입력하거나, 새로운 XML 구조를



〈그림 2〉 템플릿 관리부의 구성 및 처리도

생성하기 위해 문서의 기본 정보를 입력하면 XML 처리기의 템플릿 관리부에서 XML의 논리적인 구조정보를 분류하여 저장함에 따라 사용자 인터페이스를 이용해 구조를 편집할 수 있는 기본 정보를 생성한다. 저장된 정보를 바탕으로 사용자는 각각의 인터페이스를 통하여 문서의 구조정보를 직접 편집할 수 있고 편집된 정보는 문서 검증부를 통하여 새로운 구조를 가진 XML 문서로 저장한다. 만약 사용될 문서에 수학적 표현이 필요하다면 XML 문서 편집 중에 본 시스템이 포함하는 MathML 처리기를 통해 MathML의 구조를 자동으로 생성한다. XML 문서 내에 포함된 수식구조는 웹 환경에서 사용자가 읽거나 보기 쉽도록

ActiveX 처리기를 이용하여 문서의 가독성을 향상시킨다.

### 3.1 XML 문서 처리기

#### 3.1.1 템플릿 관리부

XML 템플릿 관리부는 사용자로부터 입력받은 XML 문서나 DTD 문서에서 문서구조를 생성할 수 있는 논리적인 구조정보를 저장하여 직접적인 문서편집에 필요한 기본정보를 생성한다.

〈그림 2〉는 템플릿 관리부의 구성요소 및 처리관계를 보인다. 사용자로부터 입력받은 DTD 문서나 XML 문서는 파서를 통해 문서의 정형성을 검사한 후 기본 XML 선언부의 내용인 문서 버전(version),

인코딩(encoding) 형식, 독립문서선언(stand-alone)을 수집하고 DTD 문서 내에 정의되어 있는 파라미터 엔티티(Parameter Entity)의 그룹을 생성한다. 수집된 엔티티 그룹의 내용을 통하여 시스템은 DTD 문서와 엔티티 그룹의 엔티티 선언문을 매핑한다. 매핑된 엔티티 정보들은 그것들 중에서 일반적인 엔티티(General Entity)들을 선별하여 문서 편집부의 엔티티 삽입 모듈에 사용될 논리정보를 해쉬 테이블에 저장한다. 엔티티 매핑이 완료되면 시스템은 엘리먼트 그룹과 속성 그룹을 생성한다. 엘리먼트 그룹과 속성그룹은 XML의 논리적인 구조를 이들 엘리먼트들과 각 엘리먼트들이 포함하는 속성들을 수집하게 되고 이렇게 수집된 정보도 마찬가지로 해쉬 테이블에 저장한다.

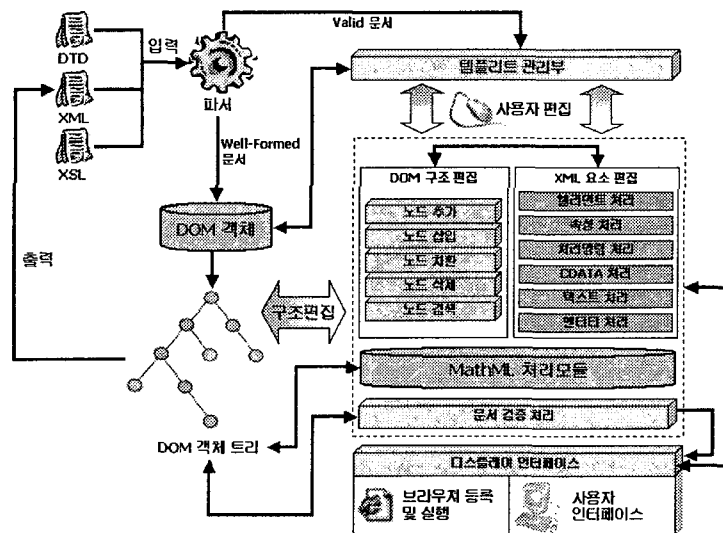
기본 논리 정보들이 해쉬 테이블에 저

장이 완료되면 시스템은 XML 구조의 기본 골격을 이들 템플릿을 구성한다. 템플릿 구성은 XML의 각 구성요소들을 해쉬 테이블로부터 발췌하여 논리구조에 맞게 각각의 구성요소들의 포함관계를 최소 템플릿나 원본 XML의 논리구조에 일치시켜 DOM(Document Object Model) 객체 트리를 생성한다(W3C 2002. <http://www.w3.org/DOM>).

### 3.1.2 XML 문서 편집부

XML 문서 편집부는 템플릿 관리부에서 생성된 DOM 객체 트리를 각각의 사용자 인터페이스를 통해 XML 논리구조에 접근하여 이를 편집하게 한다.

<그림 3>은 XML 문서 편집부의 구성 요소 및 처리 관계를 보인다. XML 문서



<그림 3> XML 문서 편집부의 구성 및 처리도

편집부의 설계는 크게 Valid 문서와 Well-Formed 문서 처리의 두 가지 방식으로 설계되었다.

첫째로, 사용자로부터 입력받은 문서가 DTD를 참조하거나 DTD 문서일 경우에는 Valid 문서로 판단하여 문서를 템플릿 관리부에 넘겨준다. 템플릿 관리부에 저장된 XML 논리구조 정보를 이용하여 DOM 객체 트리를 생성하게 되고 생성된 DOM 객체 트리를 이용해 사용자가 쉽게 논리구조를 편집할 수 있도록 CTreeCtrl 클래스를 상속받는 CNode-TreeView 클래스에 의해 트리 구조 인터페이스를 생성하고, 한편 XML 문서 소스 자체를 다룰 수 있기 위해서 CRichEditCtrl 클래스를 상속받는 CTextEditView 클래스에 의해 XML 소스 중에 PCDATA 노드만을 편집할 수 있는 소스편집 인터페이스를 생성하여 일반 사용자들이 직접 소스를 편집함에 따른 오류를 최소화시킨다. 한편, 일반적인 엔티티(General Entity) 들을 문서 구조상에 직접 삽입하기 위해 CListCtrl 클래스를 상속받는 CListEntity 클래스에 의해 트리 인터페이스와 소스편집 인터페이스의 PCDATA 사이에 해당 엔티티를 삽입할 수 있는 인터페이스를 생성한다

(Microsoft, <http://msdn.microsoft.com/xml/default.asp>).

### 3.1.3 XML 문서 검증부

각각의 인터페이스를 통해 DOM 객체의

논리구조에 직접 접근하여 문서를 편집할 수 있게 되면서, 사용자는 자신이 편집한 논리구조가 올바르게 작성된 구조인지를 확인하기 위하여 문서검증 처리를 통해 오류를 판단하고, 메시지 창을 통해 검증된 결과를 확인할 수 있다. 문서검증 처리는 파서가 제공하는 get\_parseError 메소드를 이용하여 문서 구조의 유효성 및 정형성을 검사한다.

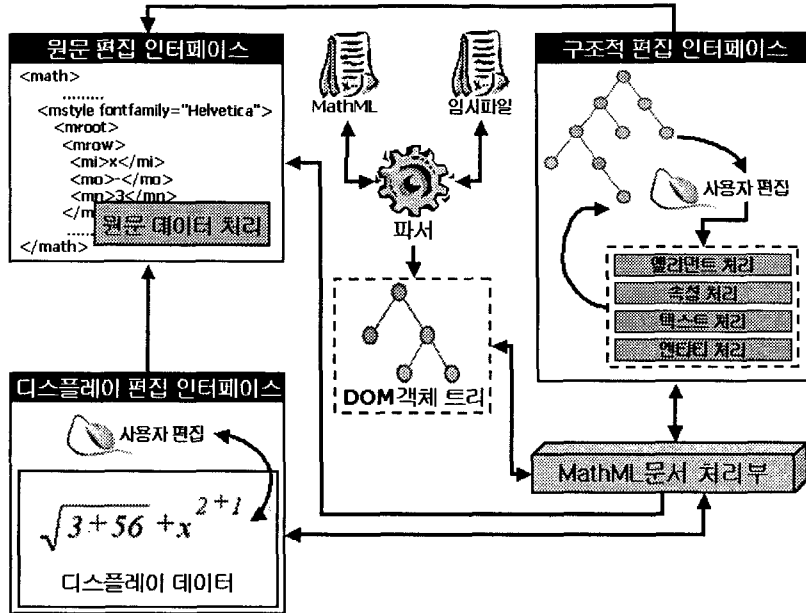
## 3.2 MathML 문서 처리기

### 3.2.1 MathML 문서 편집부

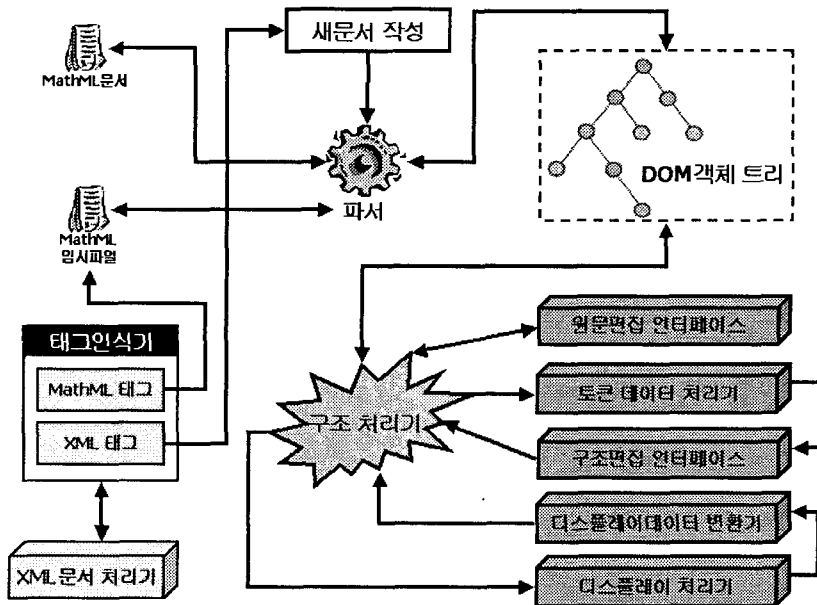
문서 편집부에서는 MathML의 기본적인 원문 텍스트의 편집기능을 제공하는 원문 편집 인터페이스와 논리적인 계층구조를 표현하고 이를 편집할 수 있도록 해주는 구조적 편집 인터페이스, 마지막으로 MathML 문서에 맞게 수학적식이 디스플레이 되는 부분을 직접 편집할 수 있도록 해주는 디스플레이 편집 인터페이스로 이루어져 있다.

〈그림 4〉는 MathML 문서 편집부의 구성요소 및 처리 관계를 보인다. 각각의 인터페이스를 통해 처리되는 데이터는 인터페이스 특성에 따라 달라지게 되는데, 원문편집 인터페이스는 텍스트 형태의 원문데이터를 통해 작동되고, 구조적 편집 인터페이스는 구조적 데이터, 디스플레이 편집 인터페이스는 입력좌표나 크기 등과 같은 디스플레이 정보 데이터를 통해 작동한다. 편집 시스템의 특성 상, 어떠한 편집





〈그림 4〉 MathML 문서 편집부의 구성 및 처리도



〈그림 5〉 MathML 문서 처리부의 구성 및 처리도

인터페이스를 통하여 해당 데이터가 변경되었을 시에, 그 이외의 다른 인터페이스를 통해 제공되는 데이터 역시 변경된 데이터와 같은 의미를 갖는 데이터로 변경될 수 있도록 인터페이스간의 데이터 동기화가 이루어져야 한다. 이는 문서 처리부의 MathML 구조처리를 통해 이루어진다(Amaya 2002)

(<http://www.w3.org/Amaya>).

### 3.2.2 MathML 문서 처리부

MathML 문서 처리기에서 처리되는 모든 종류의 데이터는 MathML 구조처리를 통해서 처리한다.

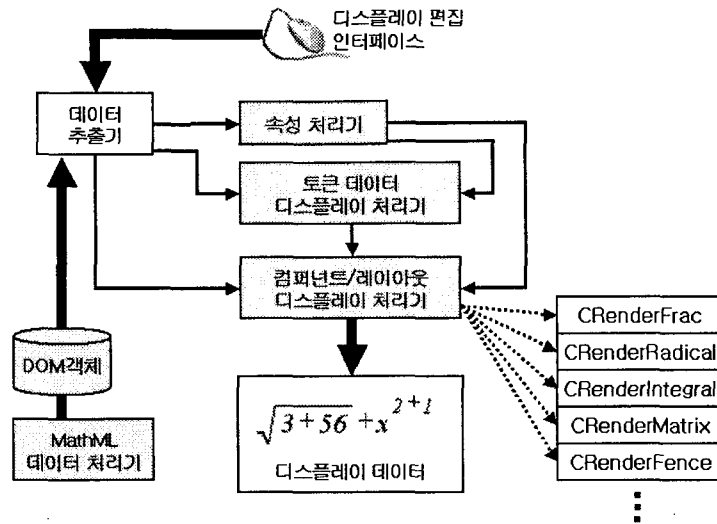
〈그림 5〉는 MathML 문서 처리부의 구성 요소 및 처리관계를 보여준다. MathML 문서 처리부는 구조 처리기의 동작이 중심이 되는데, 이는 DOM 객체를 각각의 편집 인터페이스에서 처리되는 데이터로 변환하는 처리와 반대로 편집 인터페이스에서 처리되는 데이터를 DOM 객체로 변환하는 처리로 나뉜다. 전자는 MathML 데이터 관리기에 저장되어 있는 DOM 객체가 구조 처리기를 통하여 MathML 문서의 노드 정보로 추출되고 이를 각각의 인터페이스에 쓰이는 데이터 타입에 맞게 변환하는 처리이다. 구조적 편집 인터페이스로 데이터가 전송되는 경우에는 중간에 토큰 데이터 처리기를 통하여 데이터 수정이나 삽입 시에 올 수 있는 토큰 데이터 그룹의 입력을 처리한다. 후자의 경우는, 편집을 통해 변경된 하나의 인터페

이스 데이터를 다른 인터페이스에 적용하기 위한 초기 과정이라 할 수 있다. 이 과정에서 디스플레이 처리기를 통해 전달 받은 디스플레이 정보 데이터는 디스플레이 처리 특성상 MathML 문서 구조와는 다른 형태이기 때문에 디스플레이 데이터 변환기를 통해 구조 변환 처리가 이루어진 후에 구조 처리기에 전달한다.

### 3.2.3 MathML 디스플레이 처리부

수학식이 디스플레이 되고 이를 편집하는 환경을 제공하는 디스플레이 처리부의 구성은 〈그림 6〉에서 보여준다.

디스플레이 편집 인터페이스를 통해 편집하거나, 다른 인터페이스를 통해 편집한 내용이 MathML 구조처리를 통해 디스플레이 처리부로 반영될 경우, 데이터 추출기를 통해 기존의 내용 및 변경된 내용을 조합한다. 이를 수학식 디스플레이의 기본적인 분류 형태인 토큰 데이터와 컴포넌트 및 레이아웃에 관련된 데이터로 분리하여, 각각을 토큰 데이터 디스플레이 처리기와 컴포넌트/레이아웃 디스플레이 처리기로 전달한다. 토큰 데이터는 컴포넌트나 레이아웃의 하위 요소이므로 각각의 컴포넌트나 레이아웃이 디스플레이 되는 크기에 비례하여 디스플레이 되도록 하는 처리가 이루어진다. 컴포넌트/레이아웃 디스플레이 처리기는 〈그림 6〉과 같이 형태에 따라 클래스로 구분되며, 속성처리를 통해 토큰 데이터를 포함한 각각의 형태에 적용되는 속성을 인



〈그림 6〉 디스플레이 처리부의 구성 및 처리도

식하여 처리한다(Amaya 2002)  
(<http://www.w3.org/Amaya>).

### 3.3 ActiveX 처리기

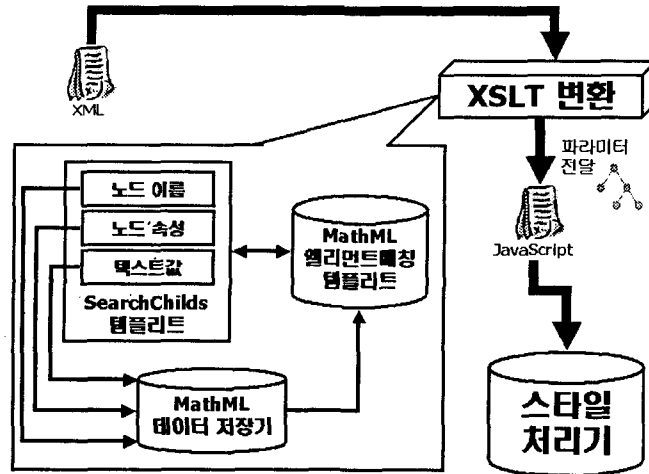
#### 3.3.1 기호 처리부

기호 처리부는 XML 문서 내에 MathML 태그가 존재하는 지를 판단하여 XSLT내에서 하나의 모듈로 동작하여 “<math>...</math>”와 같은 태그를 판단하고 처리한다. 만약 문서내 MathML 태그가 존재한다면, 기호 처리부는 태그들을 재귀순환을 통해 수집하여 MathML 논리구조를 구성한다. 구성된 논리구조는 객체 트리로 반환되고 반환된 MathML 객체의 트리를 자바스크립트 구문의 파라미터로 전달한다.

#### 3.3.2 임베드 및 배포 처리부

기호 처리부로부터 전달받은 MathML 논리구조는 자바 스크립트 파일에 의해 임베드 및 배포 처리기로 파라미터 변수를 전달하고 전달받은 MathML 논리구조는 오브젝트 임베드 방식으로 ActiveX 컨트롤을 배포한 후 적당한 위치에 임베딩한다. ActiveX 컨트롤을 임베드 하기위한 HTML구문의 사용은 아래와 같다.

```
<object classid = "CLSID:EA67B6AE-D764-4734-ADC8-E37C10749278" codebase = "MathAct.CAB # version = 1,0,0,1" width="400" height="330" id="MathAct">
<PARAM NAME="FileName" VALUE="<math>.....</math>" />
</object>
```



〈그림 7〉 엘리먼트 처리부의 구성 및 처리도

위의 구문에서 classid는 본 시스템의 ActiveX 처리기에 해당하는 고유한 식별자를 기술한 것이고, codebase는 본 시스템을 배포하는 파일명이나 사이트 명을 기술하며, version은 버전 정보를 나타낸다. 또한 <PARAM> 요소의 사용은 ActiveX 처리기에 넘겨줄 파라미터를 기

술하는 것으로 NAME 속성에 브라우저 시킬 파일명을 기술하고, VALUE에 포매팅처리를 위한 MathML 태그를 넘겨준다. <표 1>은 오브젝트 임베드 방식의 객체 정의를 위한 설명이다(Frank Boumphrey 1998).

〈표 1〉 오브젝트 임베드 요소의 정의

요소명	속 성	설 명	비 고
object	classid	고유한 ActiveX 컨트롤의 식별자를 나타낸다.	수정불가
	codebase	ActiveX 컨트롤을 배포하는 배포 사이트를 기술한다.	수정불가
	version	프로그램을 작성하여 배포할 당시의 파일 버전을 나타낸다.	버전확인
	width	프로그램이 실행될 웹 브라우저 내에서의 가로 크기를 픽셀 단위로 나타낸다.	가로길이
	height	프로그램이 실행될 웹 브라우저 내에서의 세로 크기를 픽셀 단위로 나타낸다.	세로길이
	param	플러그인 방식의 HTML 코드인 EMBED 문장에 사용하던 매개변수를 지정한다.	
	value	작성한 파일명을 기입한다.	

### 3.3.3 엘리먼트 처리부

MathML 엘리먼트 처리부는 XML 문서에서 모든 MathML 엘리먼트를 찾아서 그 하위 노드들의 이름, 속성, 텍스트 값을 순서대로 재구성하여 ActiveX 처리기에 파라미터 값으로 넘긴다. <그림 7>은 엘리먼트 처리부의 구성요소 및 처리관계를 보인다(W3C 2000, <http://www.w3.org/TR/xsl>, W3C 1999, <http://www.w3.org/TR/xslt>, W3C 1999, <http://www.w3.org/TR/xpath>).

MathML 엘리먼트 처리기는 XSLT와 자바 스크립트 구문을 혼합하여 사용한다. MathML 엘리먼트 매칭 템플리트는 XML문서의 모든 MathML 엘리먼트를 매칭시킨다. 이 템플리트는 또한, SearchChilds 템플리트를 호출함으로써 MathML 엘리먼트의 모든 하위노드와 형제노드, 속성노드, 텍스트 노드를 재구성한다. SearchChilds 템플리트는 재귀순환을 통하여 재구성한 노드를 MathML 데이터 저장기에 저장하여 최종적으로 ActiveX 처리기에 넘겨준다.

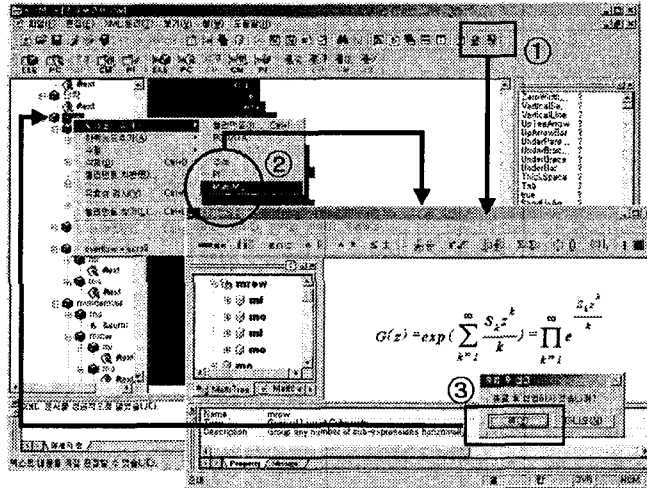
## 4 구 현

본 시스템은 IBM-PC 호환 컴퓨터 (Pentium III-800이상) Windows 2000 Server 운영체제 환경에서 Microsoft Visual C++ 6.0에 Service pack 5를 설치, 사용하여 구현하였다.

### 4.1 XML 문서 처리기 및 MathML 문서 처리기

XML 문서 처리기는 데이터를 처리 및 관리하기 위한 파서로 MSXML4.0을 사용하였고 특히, 템플리트 관리부는 Share DLL(Dynamic Linking Language)로 개발하여 다른 시스템에서도 재사용이 가능하도록 구현하였다. 아래의 <그림 8>는 XML 문서 처리기 및 MathML 문서 처리기의 구현 화면이다. XML 문서 처리기의 상단에 있는 인터페이스들은 편집에 필요한 단축 아이콘들을 나타내고 화면 좌측의 인터페이스는 XML 문서의 논리 구조를 트리 형태로 보여줌으로써 구조를 한눈에 알아볼 수 있게 하였다. 화면의 중앙에 있는 인터페이스는 XML 문서의 원문 데이터를 각 노드마다 지정된 색상으로 표현하여 보여줌으로써 사용자가 원문 데이터에서 PCDATA 노드를 구분하여 편집 할 수 있도록 하였다. 또한 화면 우측의 인터페이스는 XML 문서에 선언되어 있는 엔티티들을 리스트형태로 보여주고 원문 데이터의 PCDATA 노드 사이에 편집 할 수 있게 하였다. 마지막으로 화면 하단에 있는 인터페이스는 문서 검증부를 통해 검증된 결과를 메시지 창으로 보여줌으로써 사용자가 편집한 구조가 올바르게 작성되었는지를 확인하고 수정 할 수 있게 하였다.

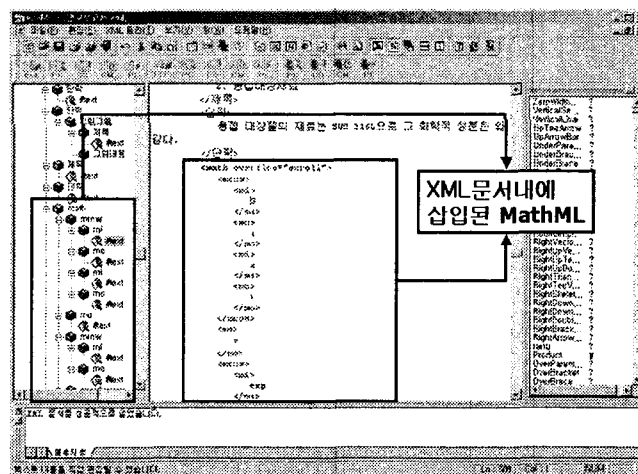
<그림 8>의 우측 하단에 있는 MathML 문서 처리기는 문서를 파싱하여 문서에서 선언한 내용에 따라 원하는 결과물을



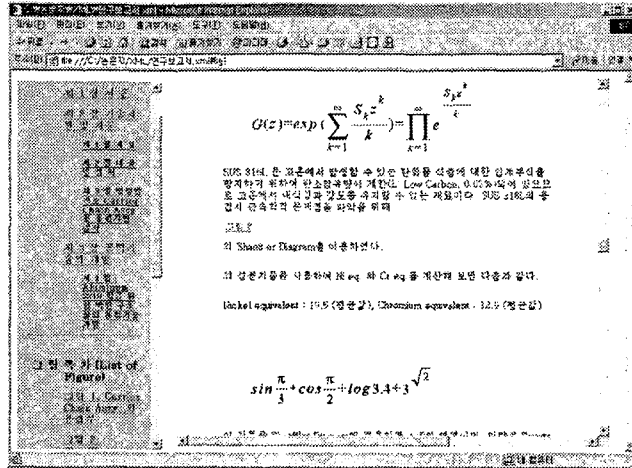
<그림 8> XML 문서 처리기 및 MathML 문서 처리기의 구현

DOM 형태의 결과로 변환해 주기 위한 기존의 XML 파서를 사용하지 않고, 이를 직접 구현함으로써 속도 향상 및 외부 모듈로부터의 의존도를 최소화하였다. 화면 좌측의 인터페이스는 MathML의 논리 구조를 트리 형태로 보여줌으로써 사용자

가 구조상에 쉽게 접근하도록 하였다. 우측은 사용자가 MathML 문법을 알지 못하더라도 원하는 수학식의 표현을 직접 편집할 수 있게 해주는 디스플레이 인터페이스이다. 마지막으로 화면의 하단에 있는 인터페이스는 각각의 노드들이 가지



<그림 9> MathML 삽입 결과



<그림 10> MathML 포매팅 결과

고 있는 정보들을 화면으로 보여줌으로써 사용자가 쉽게 해당 노드를 이해할 수 있게 구현하였다. MathML을 XML 구조상에 삽입하기 위해서는 <그림 8>에 보이는 것과 같이 XML 문서 처리기의 단축아이콘을 이용하거나 단축 메뉴를 이용하여 MathML 문서 처리기를 호출한 후 사용자가 원하는 형태의 수학적 편집을 마치고 종료 버튼을 누르면, 메시지 창을 통해 MathML의 삽입여부를 확인한 후 종료가 되면 XML 문서 구조상에 MathML 태그가 삽입이 되도록 구현하였다. <그림 9>는 MathML 삽입이 완료된 결과 화면을 보인다.

#### 4.2 ActiveX 처리기

ActiveX 처리기는 shared DLL모드에서 컴파일 하였기 때문에 컨트롤 자체에 모든 MFC DLL을 포함하고 있어 별도로

배포할 때 DLL을 따로 지정해 줄 필요가 없다. <그림 10>은 XML 문서에 포함된 수학적 표현 엘리먼트를 포매팅 처리하여 브라우저 한 결과를 보인다.

XML 문서를 입력하거나 편집을 통해 갱신된 XML 노드 구조의 변경에 맞추어 수학적 뷰 영역에 브라우징 해주는 역할을 한다. 모든 수학적식은 렌더링 영역에 보여지게 되며 그 과정은 MathML 포매팅 모듈을 통해 이루어진다. MathML 포매팅 모듈은 크게 토큰 포매팅 모듈과 레이아웃 포매팅 모듈로 나뉜다. 실제 화면의 구성과 배치에 관련된 레이아웃 포매팅 모듈을 통해 MathML의 구조적 정보를 화면상에 반영하고 토큰 데이터가 표현되기 위한 기반영역을 구성한다. 이러한 레이아웃 정보의 하위에는 하나이상의 토큰정보가 오게 되고 이를 토큰 포매팅 모듈을 통해 처리한다. MFOT들은 기본

적으로 MFC와 호환을 위해 CObject 클래스를 상속받아 구현하였다. 또한, XML 파서 모듈에서 구현된 XmlNode 클래스를 상속받음으로써, DOM 형태의 노드 구조에 기반 한 구조적 접근을 통해 포매팅 처리를 하도록 구현하였다(Amaya 2002. <http://www.w3.org/Amaya>, IBM Techexplorer. <http://www-3.ibm.com/software/network/techexplorer>).

## 5 고찰 및 결론

현재 기관이나 조직 내에 수많은 양의 데이터가 존재하고 있으나 대부분의 정보는 각 기관이나 조직에 따라 정형화된 형태로 남아있는 실정이다. 이렇게 제대로 관리되지 않는 정보들은 필요한 사람에게 적시에 원하는 형태로 제공되지 않는다. 이런 단점을 극복하고자 지식정보자원관리라는 새로운 개념이 도입되었으며, 축적된 지식들을 공유 및 관리하기 위한 지식정보자원의 디지털화가 실행되고 있다. 특히 정부에서는 과학기술이나 교육학술 분야 등에 중점적으로 지식정보자원관리 체제를 구축하도록 하였으며, 이들 분야에서 사용되는 정보들을 전자적인 문서로 처리하기 위한 시스템들이 구축되었다.

하지만, 기존의 전자문서처리 시스템들은 비구조적인 문서구조를 다루게 되고, 특히 이러한 시스템들에서 사용되는 수학적식의 처리 및 표현은 이미지나 텍스트 등으로

처리됨에 따라 사용자가 읽고 표현하기가 어려울 뿐 아니라 문서의 처리와 교환에 상대적인 불편을 초래하고 검색과 인덱싱 그리고 다른 시스템에서의 재사용성에 제한을 받게 된다. 또한 기존의 전자문서처리는 특정 시스템들에 종속되어 시스템에 따라 포맷변환을 통한 처리가 필요하게 됨으로써, 이기종간의 지식정보의 교환에 상당히 난해함을 보이고 있다.

그러므로, 이들 간에 정보를 공통적으로 처리하거나 공유하기 위한 데이터들을 XML로 구조화시키려는 움직임이 일기 시작하였고, 각각의 분야별로 처리되는 전자문서상에 수학적식을 표현하기 위한 방법으로 MathML을 사용하게 되었다. 하지만 올바로 된 MathML 문서를 작성하기 위해서는 이에 따른 여러 가지 전문적인 지식들이 요구되고 이러한 지식들을 바탕으로 한 문서작성 과정이 필수적이다.

이에 본 논문에서는 XML에 관한 기초 기술 연구 및 구조화된 XML문서를 검증하고, 처리 및 편집하기 위한 시스템을 설계 및 구현하여 XML문서를 지식정보자원관리시스템에서 사용하거나 조직간 지식기반의 XML 문서를 상호 교환할 수 있도록 하였다. 또 구조화된 문서상에 존재하는 수학적식을 처리하기 위해 MathML을 편집할 수 있는 시스템을 설계 및 구현함에 따라, 웹 상에서 수식 표현을 위한 MathML의 구조적 정보를 통해 MathML 문서를 포매팅 처리하여 제공하



고, 수식 표현을 위한 MathML 문서를 효율적으로 표현함으로써 웹 상에서 수식을 포함한 많은 지식정보자원의 표현이 가능하여 많은 관련 분야에 많이 활용될 수 있을 것이며, XML과 MathML 사용의 보편화에 크게 기여할 것이라 사료된다.

추후 본 연구는 XML 기술적인 측면을 처리하는 네트워크 상에 존재하는 자원을 참조하는 좀더 광역적인 연구나, 지식정보자원분야의 관련 화학식을 표현할 수 있는 CML(Chemical Markup Language) 등의 여러 XML 관련 어플리케이션과 XML을 연동 할 수 있는 연구가 계속되어야 할 것이다.

## 참 고 문 헌

- 정회경, 홍성찬, 이수연. 2001. 『熱血講義 XML by Example』. 이한 디지털리.
- 한국전산원. 2001. 『지식정보자원관리 표준화 및 기술개발 방안 연구』. [서울]: 한국전산원.
- Richard Anderson 외. 2000. *Professional XML*. Worx Press. 145-183.
- Frank Boumphrey. 1998. *Professional Style Sheets for HTML and XML*. Worx Press.
- W3C. 1998. Extensible Markup Language (XML) Version 1.0. <<http://www.w3.org/TR/REC-xml>>.
- W3C. 2001: Mathematical Markup Language Ver 2.0: <<http://www.w3.org/TR/MathML2>>.
- W3C. 2002. Document Object Model. <<http://www.w3.org/DOM/>>.
- W3C. 2000. "Extensible Stylesheet Language (XSL) Version 1.0". <<http://www.w3.org/TR/xsl>>.
- W3C. 1999. XSL Transformations (XSLT) Version 1.0. <<http://www.w3.org/TR/xslt>>.
- W3C. 1999. XML Path Language (XPath) Version 1.0. <<http://www.w3.org/TR/xpath>>.
- IBM Techexplorer. <<http://www-3.ibm.com/software/network/techexplorer>>.
- Amaya - W3C's Editor/Browser. 2002. <<http://www.w3.org/Amaya>>.
- Microsoft. MSDN Online XML Developer Center. <<http://msdn.microsoft.com/xml/default.asp>>.