

XML을 이용한 철근콘크리트 건물 구조계산서 전자화

XML-Based Digitalization of Structural Design Sheets for RC Buildings

정 종 현†

Jung, Jong-Hyun

강 경 수*

Kang Kyung-Soo

(논문접수일 : 2005년 2월 12일 ; 심사종료일 : 2005년 6월 10일)

요 지

본 연구의 목적은 XML을 기반으로 하여 웹(web)을 통한 교환이 가능한 철근콘크리트 건물의 구조계산서를 전자화(digitalization)하는 방안을 제시하는 것이다. 이를 위해서 철근콘크리트 건물의 구조계산서를 XML로 표현하기 위한 자료 구조(data structure)를 정의하였다. 여기에는 수식과 그래픽 등 웹에서는 교환하기 어려운 형식의 자료들도 포함된다. 다음에는 이 자료구조에 따라 구조계산서를 XML문서로 작성하였다. 그리고 이 XML문서를 웹에서 효과적으로 출력(presentation)하는 방안을 도출하였다. 마지막으로 XML로 표현된 구조계산서를 웹에서 교환할 수 있는 웹 응용프로그램(web application)을 부분적으로 구현하였고, 이를 간단한 예제에 적용하여 본 연구결과의 타당성을 검토하였다.

핵심용어 : 구조계산서, XML, 전자화

Abstract

This study describes the XML-based digitalization of structural design sheets for RC buildings to exchange on the web. For this purpose, first, the data structure of XML document that represents the structural design sheets, including mathematical expressions and graphics that cannot be easily exchanged on the web, is defined. Then, the presentation of the XML documents on the web is discussed. The prototype that facilitates the web-based exchange of the XML documents are developed and the feasibility of the results of this study is discussed.

keywords : structural design sheets, XML, digitalization

1. 서 론

1.1 연구의 배경

구조계산서(structural design sheets)는 구조해석과 부재설계를 수행하는 과정에서 생성되는 여러 자료와 결과물을 체계적으로 정리하여 기록한 것이다. 그러므로 구조계산서는 건물의 안전을 확보하기 위한 구조설계자의 설계의도, 방법, 그 결과 등이 일목요연하고 구체적으로 기록된 문서(document)이다. 이 구조계산서는 시공에 필요한 구조도면과 제작도면(shop drawing)의 작성, 시공과정에서 발생하는 구조적인 문제의 해결뿐 아니라, 설계변경, 안전진단, 증축 및 개축을 위한 구조검토 등에도 사용된다. 따라서 구조계산서는 건축도면이

나 구조도면과 마찬가지로 많은 관련자들과 교환되어야 하는 중요한 문서이다.

그러나 현재는 구조계산서를 종이에 출력하여 보관하고 그 양이 적어도 수백 쪽에 이를 정도로 매우 많기 때문에 다른 관련자와 구조계산서를 교환하는 데에 많은 어려움이 있다. 이에 따라 유사한 건축물에 대한 구조설계를 수행하는 과정에서 참조하기도 어렵다(김성아, 2002).

따라서 건물을 보다 효율적으로 설계 및 시공하고, 체계적으로 안전관리를 수행하기 위해서는 여러 관련자들과 구조계산서를 교환할 수 있어야 한다. 이를 위한 가장 효과적인 방법은 인터넷의 웹(web)을 통해서 여러 관련자들과 신속하고 정확하게 구조계산서를 교환할 수 있도록 전자화(digitalization)하는 것이다.

† 책임저자. 정회원 · 경남대학교 공과대학 건축학부 조교수

전화: 055-249-2681 ; Fax: 055-249-2682

E-mail: ironbell@kyungnam.ac.kr

* 정회원 · 동명정보대학교 건축공학과 조교수

• 이 논문에 대한 토론을 2006년 3월 31일까지 본 학회에 보내주시면 2006년 6월호에 그 결과를 게재하겠습니다.

TRANSFER BEAM B TYPE (34FY)		연말연시 시공일	DATE
		MR - 101, 103, 105, 108C	NO. /
		fc = 400 kg/cm ² ; fy = 4000 kg/cm ²	
TG1 :	600 X 2000	TG2 :	400 X 2000

그림 1 보의 설계결과를 나타내는 구조계산서의 일부

1.2 연구의 목적

구조계산서는 구조해석과 부재설계의 내용을 포함하기 때문에 그에 포함되어 있는 자료들의 양이 상당히 많고 서로 복잡한 관계를 갖고 있다. 또한 그림 1에 보이는 것처럼 문자(text)뿐만 아니라 표(table), 그래픽(graphics), 수식(mathematical expressions), 이미지(image) 등 다양한 형태를 갖는다. 이러한 이유 때문에 구조계산서를 신속하고 정확하게 웹에서 교환하기 위한 전자화에 어려움이 있다.

현재 웹에서는 주로 HyperText Markup Language(이하 HTML)를 이용하여 문서를 교환하고 있다. 하지만 HTML은 웹에 문서를 출력(presentation)하는 방법, 즉 글꼴의 종류, 크기, 정렬방식 등을 규정하는 것이기 때문에 문서에 포함되어 있는 자료의 의미(semantics)를 전달하기에는 부적합하다. 또한 문자, 표, 이미지 외에 수식과 그래픽 등을 출력하기도 어렵다. 이에 따라 HTML의 대안으로서 eXtensible Markup Language(이하 XML)가 개발되었다. XML은 문서에 포함되어 있는 자료의 내용과 의미를 웹을 통하여 전달하기 위한 것으로 단순하면서도 확장성을 갖고 있다. 또한 특정 하드웨어(hardware)나 소프트웨어(software)의 환경에 의존하지 않는다. 이러한 장점 때문에 현재 정보기술 분야에서 크게 주목받고 있으며, 건설 CALS/EC의 전자문서 표준에 해당한다(건설교통부, 2003).

본 연구에서는, 이러한 점을 고려하여, 구조계산서를 XML 문서로 작성하여 웹에서 교환할 수 있도록 구조계산서에 포함되어 있는 양이 많고 복잡한 관계를 갖는 다양한 형태의 자료들을 체계화하여 자료구조(data structure)를 정의하고, 이에 따라 작성한 XML문서를 웹에 적절히 출력할 수 있는 방안을 찾고자 한다.

1.3 연구의 내용과 범위

본 연구는 다음과 같은 내용으로 진행된다.

첫째, 구조계산서에 포함되어 있는 수식과 부재의 단면 설계 결과 등을 나타내는 그래픽을 XML로 표현하여 웹에서 전달하고 출력할 수 있는 방안을 도출한다.

둘째, 구조계산서를 XML로 표현하기 위해서 필요한 자료들을 도출하고 그 관계를 설정하여 자료구조를 정의한다.

셋째, XML문서는 그 자체로는 웹에서 어떻게 출력하는지에 대한 자료가 포함되어 있지 않기 때문에 이를 위한 별도의 여러 방법이 존재한다. 본 연구에서는 구조계산서에 적합한 출력방안을 도출한다.

넷째, XML문서로 표현된 구조계산서를 웹(web)에서 문자, 표, 이미지, 수식, 그래픽 등이 포함된 실제 구조계산서 형태로 교환할 수 있는 웹 응용프로그램(web application)을 부분적으로 개발하고 본 연구 결과의 타당성을 평가한다.

구조계산서는 구조물의 용도나 재료에 따라 매우 다양한 구성을 갖기 때문에 그 범위가 매우 넓다. 그러므로 본 연구에서는 국내에서 많이 지어지는 중·저층 철근콘크리트 건물의 구조계산서를 연구범위로 한다.

1.4 국내의 건설분야의 XML 관련 연구현황

국내에서는 우선 이주영 등(2003)과 이재영 등(2004)이 건설분야의 설계정보를 XML로 표현하는 방법을 제안하였다. 이 중, 이주영 등의 연구에서는 국내 건설분야 2차원 설계정보인 KOSDIC을 XML문서를 변환하여 웹에서 교환할 수 있는 프로그램의 원형(prototype)을 개발하였다. 김성아(2002)는 시설물관리를 위해서 건물의 공간과 공간을 구분하는 구조물을 XML로 표현하는 방법으로서 BSDL을 제안하였다. 그리고 아파트를 대상으로 일부를 구현하였다.

해외에서는 Zhu 등(2001)이 사용자가 웹을 통하여 자신의 필요에 따라 원하는 자료를 얻을 수 있는 'Malleable Frame'을 제안하였고 이를 바탕으로 XML을 이용하여 웹기반의 통합에 대한 가능성을 보여주었다. Kerrigan 등(2005)은 XML을 기반으로 법규 검토 시스템을 개발하였는데, 법규 검토를 하기 위해서 필요한 관련 법규를 XML로 표현하는 방안을 제시하였다.

이상의 연구들에서는 대체로 XML을 기반으로 웹에서 건설분야에 관련되는 자료들을 교환하는 방안들이 제시되었으며, 그 가능성을 충분히 보여준 것으로 평가된다. 하지만 그 목적이나 대상이 문서 자체가 아니라 문서 자체에 포함될 수 있는 자료들 혹은 도면이므로, 구조계산서와 같이 시각적 출력 형식이 일정한 중요성을 갖거나 수식 혹은 그래픽을 포함하는 경우에 효과적으로 적용할 수 있는 방안은 제시되어 있지 않다.

2. XML을 이용한 수식과 그래픽

2.1 MathML을 이용한 수식

MathML은 웹이나 일반 소프트웨어에서 수학에 관련된 내용을 편리하게 사용할 수 있도록 개발된 것으로서 XML을 이용하여 수식을 처리하는 표준이다. 그림 2는 수식 $ax^2 + x + c = 0$ 을 표현한 MathML문서와 이를 웹에서 출력한 예이다. 그림 2에 나타난 바와 같이 MathML은 태그(tag)를 이용하는 일반적인 XML문서와 동일하다. 태그는 ‘<’와 ‘>’로 둘러싸이는데, ‘mi’는 식별자(identifier)로서 변수에, ‘mn’은 숫자(number)에, ‘mo’는 연산자(operator)에 사용한다. 그리고 ‘msup’는 위첨자에 사용한다.

이러한 MathML을 적용하면 XML 기반으로 구조계산서에 포함되어 있는 수식을 효과적으로 처리할 수 있다. 여기에서 MathML을 적용하는 방법으로는 구조계산서를 표현하는 XML문서에 수식의 MathML 전체를 포함시키는 방법과 해당 MathML을 재구성할 수 있는 자료를 포함시키는 방법이 있다. 그림 2의 예에서 보면 그림 2의 MathML을 포함시키는 방법과 ‘a’, ‘x’, ‘c’, ‘0’ 등을 포함시키는 방법이다.

구조계산서에 포함되는 수식은 각 부재의 설계식이나 관련 단위 등으로 대부분 이미 결정되어 있으므로, 이에 해당되는 MathML을 미리 작성하여 XML문서에 포함시키는 것이 효과적이다. 따라서 본 연구에서는 전자의 방법을 적용한다.

2.2 VML을 이용한 그래픽

Vector Markup Language(이하 VML)는 XML을 기반으로 웹에서 vector 방식의 그래픽을 이용하기 위한 것이다. 그림 3은 일례로 붉은 색의 직사각형에 대한 VML과 그 출력 결과이다. 그림 3의 VML에는 직사각형을 나타내는 ‘rect’ 태그를 사용하였고, 그 속성으로 폭, 높이, 색에 대한

여 ‘width’, ‘height’, ‘fillcolor’ 등이 있다.

구조계산서에 포함되는 그래픽은 각 보 단면의 설계 결과와 같이 그 수도 매우 많고 각각이 모두 다르므로 그에 해당되는 VML을 모두 미리 작성하여 XML문서에 포함시키는 것은, 수식의 경우와는 달리, 비효율적이다. 따라서 본 연구에서는 구조계산서를 나타내는 XML문서에 그래픽을 표현하는 VML을 직접 추가하지 않고 해당 VML을 재구성하기 위해서 필요한 자료들만 포함시킨다. 예를 들면 보 단면에 대해서 그림 3에 보이는 것과 같은 VML을 추가하는 것이 아니라 보 단면의 폭과 축 등을 XML문서에 포함시키는 것이다. 그러면 나중에 XML문서에 있는 보의 폭과 축 등으로부터 보 단면의 출력에 필요한 VML을 재구성하고 이를 웹에 출력할 수 있다. 이 방식은 구조계산서에는 비교적 단순한 2차원 그래픽들이 포함되기 때문에 가능하며, 나중에 VML이 변경되는 경우에 효과적으로 대비할 수 있고, XML 문서에 포함되는 자료의 양도 상당히 줄어든다.

3. XML을 이용한 구조계산서의 표현

3.1 구조계산서 자료구조의 전체적인 구성

전자화를 염두에 두고 객체지향설계법(object-oriented design method)(Rumbaugh 등, 1991)에 따라 구조계산서를 분석하면, 구조계산서는 하나의 문서로서 여러 개의 장(chapter)을 포함하고, 다시 각 장은 여러 절(section)을, 다시 각 절은 여러 항(subsection)을 포함하는 집합관계(aggregation)가 있음을 알 수 있다. 물론 각 항은 다시 더 구체적으로 분석될 수 있다. 여기에 다시 각 장, 절, 항 등에는 제목(title)이 포함되어 있다는 것을 알 수 있다. 이에 따르면 ‘구조계산서’, ‘장’, ‘절’, ‘항’, ‘제목’ 등의 객체(object)를 도출하고, 이 들 사이에 집합관계를 설정할 수 있다.

하지만 이러한 분석 결과는 구조계산서의 각 부분이 어떤 내용과 형식으로 이루어져 있는지를 제대로 반영하지 못한

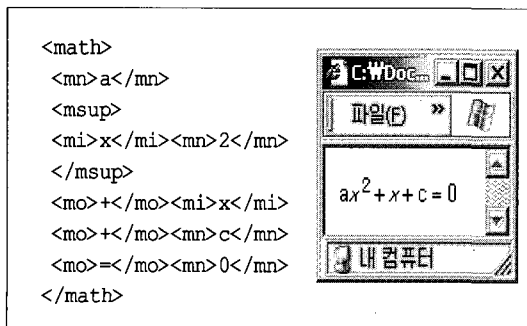


그림 2 MathML을 이용한 수식의 예

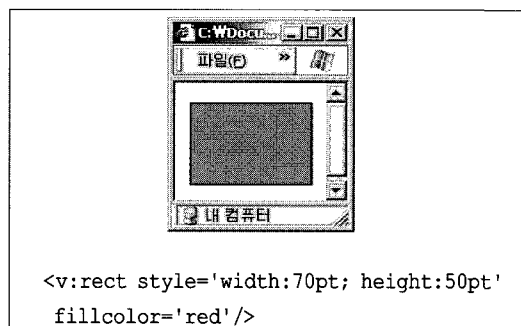


그림 3 VML을 이용한 그래픽의 예

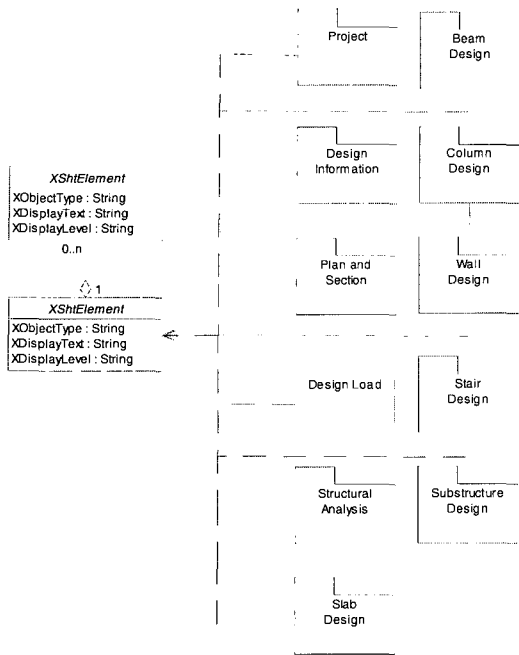


그림 4 구조계산서의 전체적인 자료구조

것이다. 구조계산서를 구성하는 각 장들은 모두 제목을 갖고 있고 여러 개의 절을 포함한다는 점은 동일하지만 그 내용과 형식은 서로 상이하다. 이는 절과 항에 대해서도 마찬가지이다. 따라서 본 연구에서는 이러한 점을 효과적으로 반영하기 위해서 구조계산서를 여러 개의 장이 아니라 작은 규모의 기본적인 단위체들로 이루어진 문서로 파악한다.

이 관점에 의하면 구조계산서는 여러 단위체들로 구성되는데, 이 단위체들은 다시 필요에 따라 서로를 포함하는 집합관계를 갖는다. 그리고 이 단위체들은 모두 그에 적합한 제목을 갖고 있다. 그림 4는 이러한 점을 정의한 반영하여 구조계산서의 전체적인 자료구조를 Unified Modeling Language의 클래스도표(class diagram)로 나타낸 것이다.

그림 4에서 구조계산서를 구성하는 기본적인 단위체는 XShElement 클래스로 표현되었다. 이 클래스의 세 개의 속성(attribute) 중에서 XDisplayText는 제목을, XDisplayLevel은 이 단위체들의 구성체계, 즉 장, 절, 항 중 어떤 것인지를 표현한다. 그리고 이 단위체들의 집합관계를 나타내기 위해서 자기 자신과 일대다(1:n)의 집합관계를 갖는다. 본 연구에서 정의되는 다른 클래스들은 거의 대부분 XShElement 클래스에서 파생(derivation)되고 각각에 필요한 자료들이 추가된다. 이에 따라 각 클래스들은 기본적인 단위체로서의 성질을 유지하고 여기에 각각의 다양한 내용과 형식도 적절히 표현할 수 있다. 이렇게 되면 구조계산서에 포함되는 각 단위체는 필요에 따라 장이 될 수도 있고, 절이나 항이 될 수도 있다.

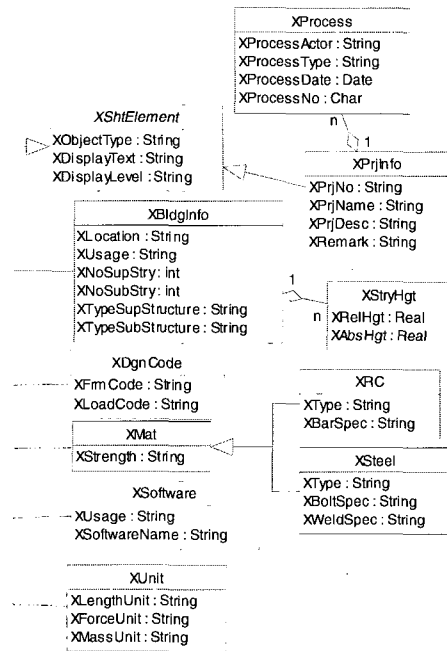


그림 5 Project와 Design Information 부분의 자료구조

그림 4에 표현된 'Project', 'Design Information' 등은 자료구조와는 직접적인 관계없이 어떤 클래스들이 정의되는지 개략적으로 분류하고 도표로 나타내기 위한 패키지(package)이다. 여기에서는 일반적인 철근콘크리트 건물의 구조계산서에 포함되는 내용을 현대건설의 구조계산서 작성 지침(현대건설, 1988)을 참고하여 패키지를 정의하였다.

3.2 구조계산서의 자료구조 상세

여기에서는 앞 절에서 기술한 전체적인 자료구조에 따라 추가적으로 정의한 자료구조 중에서 일부만 구체적으로 기술한다.

3.2.1 Project와 Design Information 부분

Project와 Design Information 패키지에서는 프로젝트와 설계 개요에 관련된 자료들을 그림 5와 같은 클래스들로 정의한다. 그림 5에서 보면 기본 단위체인 XShElement 클래스로부터 프로젝트 개요, 건물 개요, 설계기준, 재료, 사용 소프트웨어, 단위를 나타내는 XPrjInfo, XBldgInfo, XDgnCode, XMat, XSoftware, XUnit 클래스가 파생되어 구조계산서를 구성하는 단위체로 사용된다. 그리고 다시 RC와 철골을 나타내는 XRC, XSteel 클래스가 XMat 클래스에서 파생되므로, 이 클래스들 역시 기본 단위체로 사용된다. 반면에 구조설계의 진행과정과 층고를 나타내는 XProcess

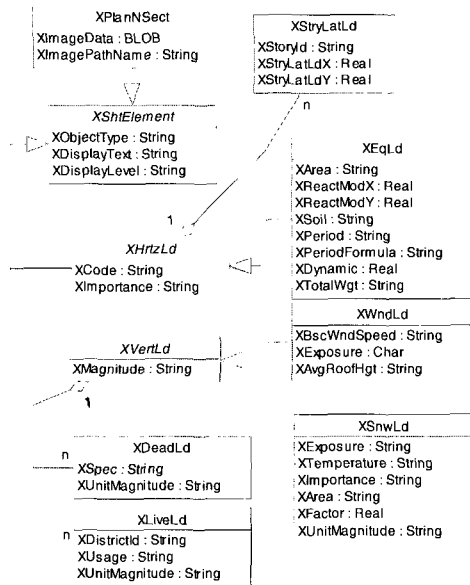


그림 6 Plan and Section과 Design Information 부분의 자료구조

와 XStryHgt 클래스는 XShtElement 클래스로부터 파생되지 않았으므로 기본적인 단위체로는 사용될 수 없다.

이 자료구조는 구조계산서를 XML문서로 작성하기 위한 것이므로, 자료구조의 속성과 자료형(data type)을 결정할 때 구조계산서에서 해당 자료의 역할과 성질을 적절하게 고려해야 한다. 일례로 XMat 클래스에 있는 XStrength는 재료의 강도임에도 불구하고 자료형을 문자열(string)로 한다. 이는 구조계산서는 강도를 단위와 함께 표현하므로 'kgf/Cm²'에 해당되는 MathML을 강도에 추가하여 함께 나타내야 하기 때문이다.

3.2.2 Plan and Section과 Design Load 부분

이 부분에는 대상 건물의 평면과 단면, 그리고 설계 하중의 상세에 관련된 자료들을 그림 6과 같이 클래스로 정의한다. 평면과 단면을 나타내는 XPlanNSect 클래스는 XShtElement 클래스로부터 파생되는데, 도면을 이미지로 나타낼 수 있도록 도면 이미지와 그 경로(path)가 속성으로 정의된다. 하중과 관련해서는 수직하중, 수평하중을 나타내는 XVertLd, XHrtzLd 클래스가 정의되며, 다시 각각으로부터 고정하중, 적재하중, 설하중, 지진하중, 바람하중을 나타내는 XDeadLd, XLiveLd, XSnwLd, XEqLd, XWndLd 클래스가 파생된다. 본 연구에서는 중·저층 건물의 구조계산서를 대상으로 하였으므로 수평하중에는 등가정적하중으로 정의하는 부분에 관련되는 자료들만 포함시켰다.

이상 3.2절에서 기술한 자료구조는 본 연구의 범위인

중·저층 철근콘크리트 건물의 구조계산서를 대상으로 정의한 것이다. 하지만, 구조계산서가 상호 집합관계를 갖는 장, 절, 항 등으로 구성된다는 점과, 각 장, 절, 항의 내용과 형식은 서로 상이하다는 점은 모든 종류의 구조계산서에서 동일하다. 그러므로 본 연구에서 XShtElement 클래스를 기본 단위체로 설정하고, 이로부터 다른 클래스들을 파생시켜 여기에 다양한 내용과 형태를 추가하는 방식은 동일하게 적용할 수 있을 것이다. 다만, 구조계산서마다 대상 건물에 따라 변화가 있으므로 구체적으로 XShtElement 클래스에서 파생시키는 클래스들의 종류와 그 구체적인 속성에는 차이가 있을 것이다.

3.3 구조계산서 XML문서

본 절에서는 앞 절에서 기술한 자료구조를 바탕으로 작성한 XML문서의 일부에 대해서 기술한다.

3.3.1 XML Schema

XML문서는 반드시 브라우저나 다른 프로그램에 의해 처리될 수 있도록 해주는 최소한의 규약과 맞는 '잘 구성된(well-formed)' 문서이어야 하고, 또한 적절한 Document Type Declaration(이하 DTD)을 포함하고 이 구조를 따르는 '유효한(valid)' 문서이어야 한다(Young, 2000). DTD에는 XML문서의 구조, 즉 태그를 이용하여 표현되는 요소(element)와 그 포함관계, 요소에 포함되는 여러 속성(attribute) 등이 정의된다. 하지만, DTD는 자료형(data type)이 제한적이고 사용하기 약간 어려운 점 등 많은 단점이 있다. 따라서 대안으로 XML Schema가 개발되었다. XML Schema는 "더 강력한 자료형 그리고 XML문서의 구조와 규칙을 기술하는 데에 있어서 더 많은 유연성을 제공하"(Whalin, 2001)는 장점이 있다. 그래서 본 연구에서는 구조계산서를 유효한 XML문서로 표현할 수 있도록 3.1절과 3.2절에서 기술한 자료구조에 따라서 구조계산서 XML Schema를 정의하였다.

그림 7은 그림 5에서 건물의 개요에 대한 자료를 나타내는 XBldgInfo 클래스와 건물의 층고를 나타내는 XStryHgt 클래스에 해당되는 부분의 XML Schema를 편집하여 나타낸 것이다. 건물의 위치, 용도, 지상 및 지하 층수, 지상 및 지하구조물의 종류는 XBldgInfo요소의 하위요소로 정의되어 있고, 각 층의 층고와 지상으로부터의 높이는 XBldgInfo요소의 하위요소인 XStryHgt 요소의 하위요소로 정의되어 있다.

```

<xsd:element name="XBldgInfo" minOccurs="0" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="XLocation" type="xsd:string" minOccurs="0" />
      <xsd:element name="XUsage" type="xsd:string" minOccurs="0" />
      <xsd:element name="XNoSupStry" type="xsd:int" minOccurs="0" />
      <xsd:element name="XNoSubStry" type="xsd:int" minOccurs="0" />
      <xsd:element name="XTypeSupStructure" type="xsd:string" minOccurs="0" />
      <xsd:element name="XTypeSubStructure" type="xsd:string" minOccurs="0" />
      <xsd:element name="XStryHgt" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="XRelHgt" type="xsd:float" minOccurs="0" />
            <xsd:element name="XAbsHgt" type="xsd:float" minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

그림 7 XBldgInfo 클래스에 대한 XML Schema

3.3.2 구조계산서 XML문서의 작성

본 연구에서는 앞 절에서 기술한 XML Schema에 맞추어 구조계산서 XML문서를 작성하였다. 구조계산서는 양이 방대하기 때문에 하나의 XML문서로 작성하면 이를 저장,

참조, 편집할 때에 많은 부하가 발생한다. 따라서 이러한 점을 고려하여 본 연구에서는 그림 4에서 패키지로 표현한 부분들을 각각의 장(chapter)으로 설정하고, 이를 단위로 하여 각 장에 해당되는 XML문서를 분리하여 따로 작성하였다.

그림 8은 설계 하중에 관련된 자료들의 XML문서이고,

```

<?xml version="1.0" standalone="yes"?>
<XDgnLdSht>
  <XDgnLdShtDisplay>
    <XObjectType>XDgnLdSht</XObjectType>
    <XDisplayText>4. 설계하중</XDisplayText>
    <XDisplayLevel>Chapter</XDisplayLevel>
  </XDgnLdShtDisplay>
  <XVertLd>
    <XVertLdDisplay> . . . </XVertLdDisplay>
    <XDeadLdList>
      <XDeadLdList...> . . . </XDeadLdList...>
      <XMagnitude>0.048 &lt; ... &gt;</XMagnitude>
      <XDeadLd>
        <XSpec>바닥마감 (3cm)</XSpec>
        <XUnitMagnitude>0.006 gt;</XUnitMagnitude>
      </XDeadLd>
      <XDeadLd> . . . </XDeadLd> . . .
    </XDeadLdList>
    <XLiveLdList>
      <XLiveLdList...> . . . </XLiveLdList...>
      <XLiveLd>
        <XDistrictId>1</XDistrictId>
        <XUsage>일반사무실</XUsage>
        <XUnitMagnitude>0.0 . . . </XUnitMagnitude>
      </XLiveLd>
    </XLiveLdList>
  </XVertLd>
  <XWndLd>
    <XWndLdDisplay> . . . </XWndLdDisplay>
    <XWndCode>KS 2000</XWndCode>
    <XImportance>(특) = 1.10</XImportance>
    <XBscWndSpeed>30 m/sec</XBscWndSpeed>
    <XExposure>A</XExposure>
    <XAvgRoofHgt>8250 cm</XAvgRoofHgt>
    <XWndStryLatLd>
      <XStoryId>Roof</XStoryId>
      <XStryLatLdX>40841.9</XStryLatLdX>
      <XStryLatLdY>20359.3</XStryLatLdY>
    </XWndStryLatLd>
    <XWndStryLatLd> . . . </XWndStryLatLd> . . .
  </XWndLd>
  <XEqlD> . . . </XEqlD>
</XDgnLdSht>

```

그림 8 설계하중 부분의 편집된 XML문서

```

<XDgnSht>
  <XBmDgnSht>
    <XBmDgnShtDisplay> . . . </XBmDgnShtDisplay>
    <XBm>
      <XName>G1</XName>
      <XMatType>Concrete</XMatType>
      <XHgt>500</XHgt>
      <XWth>300</XWth>
      <XStrength>&lt;m . . . math>></XStrength>
      <XBarStrength> . . . </XBarStrength>
      <XCover>60</XCover>
      <XBmSect>
        <XSectName> . . . </XSectName>
        <XLoc>Left</XLoc>
        <XStirrupSize>10</XStirrupSize>
        <XStirrupSpc>210</XStirrupSpc>
        <XBar>
          <XLayerNo>1</XLayerNo>
          <XBarLoc>Top</XBarLoc>
          <XBarNo>3</XBarNo>
          <XBarSize>19</XBarSize>
          <XBarCover>60</XBarCover>
        </XBar>
        <XBar> . . . </XBar> . . .
      </XBmSect>
      <XBmSect> . . . </XBmSect> . . .
    </XBm>
    <XBm> . . . </XBm> . . .
  </XBmDgnSht>
</XDgnSht>

```

그림 9 보 설계 부분의 편집된 XML 문서

그림 9는 보의 설계에 관련된 자료들의 XML문서이다. 그림 8을 보면 가장 상위에 설계하중 부분을 나타내는 <XDgnLdSht>가 있다. 이 요소는 단지 모든 XML문서에서 존재해야 하는 가장 상위의 요소를 나타내는 것으로서 그림 6에는 정의되어 있지 않다. 그 하위요소로 <XDgnLdSht>가 장(chapter)이라는 구성체와 제목을 표현하는 XSht-Element 클래스에 해당되는 <XDgnLdShtDisplay>, 바닥판 하중, 풍하중, 지지하중을 나타내는 <XVertLd>, <XWndLd>, <XEqLd> 요소가 포함된다. 다시 이 각각에는 그에 적절한 하위요소들이 포함된다. 그림 8에서 밑줄이 그어진 <XMagnitude> 요소는 하중의 크기를 나타내는데, 그 값이 0.048 kgf/cm^2 에 해당되는 MathML인 $0.048 \text{ <math . . . math>}$ 로 되어있다. 그리고 여기에서 MathML에 포함되는 태그의 시작과 끝을 나타내는 '<'와 '>'는 XML문서에서 보통 문자로는 사용할 수 없는 'reserved keyword'이므로 '<'와 '>'를 의미하는 '<:'와 '>:'로 표현하였다.

그림 9에서 보면 가장 상위에 보 설계하는 부분을 나타내는 <XBmDgnSht>가 있다. 여기에는 그 제목과 장(chapter)

으로서 역할을 한다는 것을 나타내기 위한 <XBmDgnSht-Display>, 설계 대상인 보를 나타내는 <XBm>이 포함된다. 다시 <XBm>에는 해당 속성과 철근콘크리트 보의 단면 상세인 <XBmSect> 세 개가 포함된다. 여기에는 다시 상부와 하부의 주근을 나타내는 <XBar>가 포함된다. 이 자료들은 그 자체로 구조계산서에 나타나는 것이 아니라 부재 단면을 그래픽으로 표현하는 VML을 재구성할 때 이용된다.

4. 구조계산서 XML문서의 출력

4.1 XSL을 이용한 출력

구조계산서 XML문서를 웹에 출력하는 여러 방법 중에서 우선적으로 고려할 수 있는 방법은 eXtensible Stylesheet Language(이하 XSL)를 이용하는 것이다.

XSL은 XML문서를 다른 자료구조를 갖는 XML문서나 HTML, RTF 등으로 변환할 수 있다. 따라서 XSL을 이용하면 XML문서를 HTML문서로 변환하여 웹에 출력할 수 있다. 따라서 이 방법은 문자나 표, 이미지와 같이 HTML에서 직접적으로 출력할 수 있는 형태는 쉽게 출력할 수 있지만 수식이나 그래픽 등의 형태로는 출력하기가 곤란한 단점이 있다. 그러므로 구조계산서 XML문서의 경우에는 XSL만 이용하여 웹에 출력하기는 곤란하다. 또 한 가지 단점은 구조계산서와 같이 여러 자료들을 복잡한 논리적 처리과정에 따라 출력해야 하는 경우에는 적용하기 쉽지 않다는 것이다.

4.2 DOM, XHTML, 스크립트를 이용한 출력

XML문서를 웹에 출력하는 다른 방법은 Document Object Model(이하 DOM)을 바탕으로 XHTML과 스크립트를 이용하는 것이다. 그림 10은 이 방식을 도식화하여 표현한 것이다. DOM은 XML문서를 열고 그에 포함되어 있는 여러

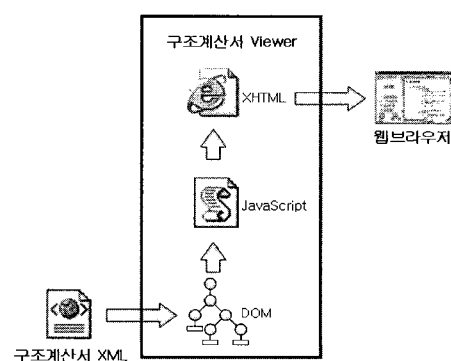


그림 10 구조계산서 XML 문서의 출력과정

```

<script language='javascript'>
  _Document = dsoDgnInfoSht.XMLDocument;
  _NodeList = _Document.getElementsByTagName("XBldgInfoDisplay");
  document.write("<ul><li>" + _NodeList(0).childNodes(2).text + "</li>");
  document.write("<ul>");
  _NodeList = _Document.getElementsByTagName("XBldgInfo");
  document.write("<li>위치 : " + _NodeList(0).childNodes(2).text + "</li>");
  document.write("<li>용도 : " + _NodeList(0).childNodes(3).text + "</li>");
  document.write("<li>지상층수 : " + _NodeList(0).childNodes(4).text + "층</li>");
  . . .
  
```

그림 11 건물 개요 부분을 출력하기 위한 JavaScript의 편집된 일부분

자료들을 간편하고 체계적으로 읽고 수정하고 추가할 수 있는 프로그래밍 모델(programming model)로서 그 유용성 때문에 많이 사용되고 있다. XHTML에서는 DOM을 이용하여 XML문서에 접근할 수 있는데, 일단 XHTML에서 DOM을 사용할 수 있게 되면 이를 XHTML내의 스크립트(script)에서 XML문서의 모든 자료를 원하는 대로 참조할 수 있다. 본 연구에서는 간단하고 호환성이 높은 JavaScript를 이용한다.

DOM을 기반으로 하여 XHTML에서 JavaScript를 이용하면 다양한 자료들이 복잡하게 얽혀 있는 XML 문서도 비교적 쉽게 처리할 수 있다. 또한, 이 자료들을 XHTML에서 직접 처리할 수 있는 형태 외에 수식이나 그래프의 형태로도 출력할 수 있다. 구조계산서의 경우에는 특히 부재 단면의 설계 결과나 풍하중 및 지진하중을 등가정적하중으로 치환한 결과를 보여주는 그래프(graph)를 그래프로 처리하는 과정에 매우 유용하게 활용할 수 있다.

5. 구조계산서 XML문서의 웹 기반 교환 사례

본 연구에서는 지상 20층의 규모의 사무실 건물을 임의로 가정하여 구조계산서 XML문서를 작성하였다. 다만, 그림 4에 표시된 패키지 중에서 'Project', 'Design Information', 'Plan and Section', 'Design Load', 'Beam Design'에 해당되는 부분만 작성하였다. XML문서는 모두 Microsoft Windows XP 환경에서 문자편집기를 이용하여 작성하였다.

5.1 구조계산서 Viewer 개발

본 연구에서는 XML로 표현된 구조계산서를 웹에서 교환할 수 있는 간단한 웹 응용프로그램을 그림 10에 나타나 있는 바와 같이 개발하였다. XML문서의 처리 과정은 4.2절에서 기술한 바와 같다. 이 웹 응용프로그램을 이용하면 구조계산서의 XML문서를 Microsoft Internet Explorer 6.0에서 읽고 출력할 수 있다.

그림 11은 본 연구에서 개발한 Viewer 중에서 건물의 개요를 출력하기 위한 JavaScript의 일부분이다. 여기에서 '_Document'가 DOM객체를 나타내는 변수이고, 밑줄이 그어진 부분이 DOM을 이용하는 부분이다. 이 중에서 'getElementsByTagName'은 'XBldgInfoDisplay'라는 이름의 요소에 포함되어 있는 XML자료를 얻는 부분이다. 그 아래의 '_NodeList(0).childNodes(2).text'는 해당 요소의 세 번째 하위요소에 저장된 건물 개요 부분의 장 제목을 얻는 부분이다. 나머지는 과 등 XHTML 요소들을 이용하여 출력하는 부분이다.

5.2 실행 결과

그림 12와 13은 구조계산서 XML문서를 웹을 통하여 교환하고 출력한 화면이다. 그림 12는 구조설계 개요로서 건물에 대한 개략적인 적인 자료들이 문자와 표의 형태로 출력된 것이다. 그림 13은 등가정적 지진하중을 산정한 결과를

층	폭(cm)	길이(cm)
1F	500	500
2F	400	500
3F	400	1300
4F	400	1700
5F	400	2100
6F	400	2500
7F	400	2900
8F	400	3300
9F	400	3700
10F	400	4100
11F	400	4500
12F	400	4900
13F	400	5300
14F	400	5700
15F	400	6100
16F	400	6500
17F	400	6900
18F	400	7300
19F	400	7700
Roof	500	8200

그림 12 구조설계 개요 출력 화면

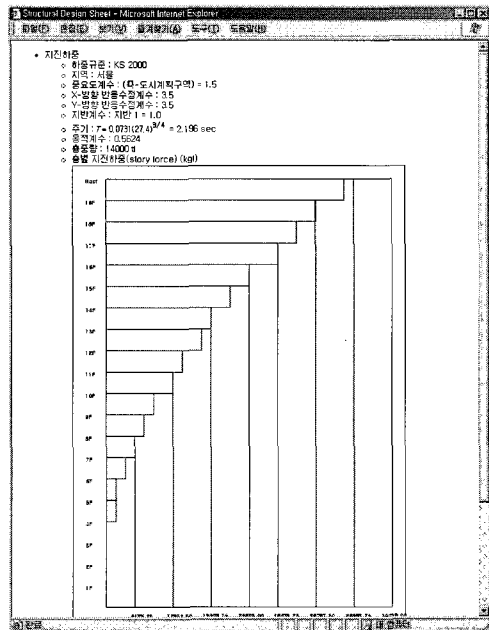


그림 13 등가정적 지진하중 산정 결과 출력 화면

나타낸 그림으로서, 그에 사용된 각종 계수들과 그 결과를 시각화한 그래프가 그래픽으로 나타난 것을 확인할 수 있다. 또한 주기를 산정하는 수식이 출력된 것도 확인할 수 있다.

6. 결 론

본 연구의 목적은 웹을 통한 교환이 가능하도록 XML을 이용하여 철근콘크리트 건물의 구조계산서를 전자화하는 것이다. 이를 위해서 XML을 기반으로 수식과 그래픽을 웹에서 처리할 수 있는 방안을 도출하였다. 그리고 구조계산서를 XML로 표현하기 위한 자료구조를 정의하였다. 다음에는 구조계산서 XML문서를 웹에서 효과적으로 출력하는 방안을 도출하였다. 마지막으로 구조계산서 XML문서를 웹에서 문자, 표, 수식, 그래픽 등이 포함된 실제 구조계산서 형태로 출력할 수 있는 간단한 웹 응용프로그램을 개발하여 본 연구 결과의 타당성을 평가하였다. 이상의 연구과정을 통하여 다음의 결론을 얻었다.

첫째, 구조계산서를 상호 적절한 집합관계를 갖는 기본적인 단위체들의 구성으로 보고, 이를 구조계산서의 자료구조에 반영해야 한다. 즉, 기본적인 단위체가 제목과 구성체계, 그리고 상호간의 집합관계를 표현하도록 정의해야 한다. 또한 해당 자료의 구조계산서에서의 역할과 성질을 고려해서 자료구조를 정의해야 한다.

둘째, 구조계산서의 수식은 MathML을 이용하되 XML 문서에 MathML을 직접 추가하는 방법이 효과적이다. 그래픽의 경우에는 VML을 이용하되, XML문서에 VML을 직접

추가하기 보다는 VML의 재구성에 필요한 자료들만 포함시키는 것이 효과적이다.

셋째, 구조계산서 XML문서에 포함되어 있는 자료들은 그 관계가 복잡하고 문자, 표, 수식, 그래픽 등 다양한 형태를 갖기 때문에, DOM을 기반으로 XHTML과 스크립트를 함께 활용하면 효과적이다.

본 연구에서는 구조계산서에 적합한 XML Schema를 정의하였고 이에 맞추어 XML문서를 작성하였는데, 향후 연구에서는 구조계산서의 효율적 관리를 위해서 구조계산서와 관련된 자료들을 데이터베이스(database)에 저장하고 이로부터 필요한 부분만을 XML문서로 생성하는 방법을 제안해야 할 것이다. 또한 건축 관련 여러 분야에서 필요한 각종 문서에도 폭넓게 적용할 수 있는 방안도 제시할 필요가 있다.

감사의 글

본 연구는 2005학년도 경남대학교 학술논문게재연구비 지원으로 이루어졌음.

참 고 문 헌

- 건설교통부(2003) 제2차 건설CALIS/EC 기본계획, p.84
 김성아(2002) 시설물관리 도구를 위한 XML 기반 건물공간 기술 방법론, *대한건축학회논문집*, 18(12), pp.71~78.
 이재영, 한치근, 김인한, 조찬원(2004) XML을 이용한 건축정보관리 시스템 설계, *한국CAD/CAM학회논문집*, 9(3), pp.238~245.
 이주영, 김인한(2003) XML기반의 건설 분야 설계정보의 표현 및 활용에 관한 연구, *대한건축학회논문집*, 19(6), pp.3~10.
 현대건설(1988) 설계실무지침 - 구조계산서 작성지침, *현대건설 해외건축사업본부 건축연구실 발생도서* 58, HDM F-0002, p.279.
 Dan, W(2001) XML for ASP.NET Developers, 김태훈, 반선화 역, 마이트Press, p.521.
 James, R., Michael, B., William, P., Frederick, E., William, L.(1991) Object-Oriented Modeling and Design, Prentice Hall, p.500.
 Michael J. Young(2000) Step by Step XML, 김용권 역, 정보문화사, p.433.
 Shawn L. K., Kinho H. L.(2005) Regulation-centric, Logic-based compliance assistance framework, *J. Comp. in Civ. Engrg.*, 19(1), pp.1~15.
 Yimin, Z., Raja, R. A. I.(2001) Web-Based construction document processing via Malleable Frame, *J. Comp. in Civ. Engrg.*, 15(3), pp.157~169.