

XML 기반의 구조계산서 전자화를 위한 자료모델

Data Model for XML-Based Digitalization of Structural Design Sheets

정 종 현*

Jung, Jong-Hyun

ABSTRACT

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

Keywords: structural design sheets, XML, digitalization, data model.

1. 서 론

건축구조물의 구조계산서(structural design sheets)는 구조해석과 부재설계를 수행하는 과정에서 생성되는 여러 자료와 결과물을 체계적으로 정리하여 기록한 문서(document)로서 효율적이고 체계적인 설계, 시공, 안전관리를 위해서 많은 관련자들과 교환해야 하는 중요한 문서이다. 그러나 현재는 구조계산서를 종이에 출력하여 보관하고 그 양이 적어도 수백 쪽에 이를 정도로 매우 많기 때문에 다른 관련자와 교환하는 데에 많은 어려움이 있다. 따라서 건설분야의 생산성 향상을 위해서는 인터넷의 웹(web)을 통하여 여러 관련자들과 신속하고 정확하게 교환할 수 있도록 구조계산서를 전자화(digitalization)가 반드시 필요하다.

하지만 구조계산서는 구조해석과 부재설계의 내용을 포함하기 때문에 자료들의 양이 상당히 많고 서로 복잡한 관계를 갖고 있다. 또한 문자(text)뿐 아니라 표(table), 이미지(image), 수식(mathematical expressions), 그래픽(graphics) 등 다양한 형태를 갖는다. 이러한 이유 때문에 현재 웹에서 문서를 교환하는 주된 방법인 HyperText Markup Language(이하 HTML) 기반으로는 구조계산서를 전자화하기 어려운 점이 많다.

이에 본 연구에서는 HTML 대신에 정보기술 분야에서 현재 크게 주목받고 있으며 건설CAIS/EC에서도 전자문서 표준으로 사용되는 eXtensible Markup Language(이하 XML)를 기반으로 구조계산서를 전자화할 수 있도록 구조계산서에 포함되어 있는 양이 많고 복잡한 관계를 갖는 다양한 형태의 자료들을 체계화하여 자료모델(data model)을 정의하였다. 그리고 이 자료모델에 따라 구조계산서를 데이터베이스(database)에 저장한 후에 XML 문서로 작성하였고 웹에서 문자, 표, 이미지, 수식, 그래픽 등이 포함된 구조계산서 형태로 교환할 수 있는 웹 응용프로그램(web application)을 부분적으로 개발하여 본 연구결과의 타당성을 평가하였다.

* 정희원 · 경남대학교 건축학부 조교수 E-mail: ironbell@kyungnam.ac.kr

2. XML을 이용한 수식과 그래픽

2.1. MathML을 이용한 수식

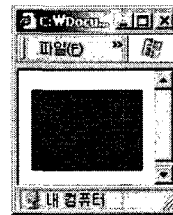
MathML은 웹이나 일반 소프트웨어에서 수학에 관련된 내용을 편리하게 사용할 수 있도록 개발된 것으로서 XML을 이용하여 수식을 처리하는 표준이다. 그림 1은 수식 $ax^2+x+c=0$ 을 표현한 MathML문서와 이를 웹에서 출력한 예이다. 이러한 MathML을 적용하면 XML 기반으로 구조계산서에 포함되어 있는 수식을 효과적으로 처리할 수 있다.

MathML을 적용하는 방법으로는 구조계산서를 표현하는 XML문서에 수식의 MathML 전체를 포함시키는 방법과 해당 MathML을 재구성할 수 있는 자료를 포함시키는 방법이 있다. 그림 1에서 보면 그림 1의 MathML을 포함시키는 방법과 'a', 'x', 'c', '0' 등을 포함시키는 방법이다. 구조계산서에 포함되는 수식은 각 부재의 설계식이나 관련 단위 등으로 대부분 이미 결정되어 있으므로, 이에 해당하는 MathML을 미리 작성하여 XML 문서에 포함시키는 것이 효과적이다. 따라서 본 연구에서는 전자의 방법을 적용한다.

```
<math>
<mn>a</mn>
<msup>
  <mi>x</mi><mn>2</mn>
</msup>
<mo>+</mo><mi>x</mi>
<mo>+</mo><mn>c</mn>
<mo>=</mo><mn>0</mn>
</math>
```



그림 1. MathML을 이용한 수식의 예



```
<v:rect style='width:70pt; height:50pt'
fillcolor='red' />
```

그림 2. VML을 이용한 그래픽의 예

2.2 VML을 이용한 그래픽

Vector Markup Language(이하 VML)는 XML을 기반으로 웹에서 vector 방식의 그래픽을 이용하기 위한 것이다. 그림 2의 예는 붉은 색의 직사각형에 대한 VML과 그 출력 결과이다. 그림 2의 VML에는 직사각형을 나타내는 'rect' 태그를 사용하였고, 속성으로 폭, 높이, 색을 나타내는 'width', 'height', 'fillcolor'가 있다.

구조계산서에 포함되는 그래픽은 각 보 단면의 설계 결과와 같이 그 수도 매우 많고 각각이 모두 다르므로 그에 해당하는 VML을 모두 미리 작성하여 XML문서에 포함시키는 것은, 수식의 경우와는 달리, 비효율적이다. 따라서 본 연구에서는 구조계산서를 나타내는 XML문서에 해당 VML을 재구성하기 위해서 필요한 자료들만 포함시킨다. 예를 들면 보 단면에 대해서 그림 2에 보이는 것과 같은 VML을 추가하는 것이 아니라 보 단면의 폭과 춤 등에 해당하는 자료들을 XML문서에 포함시키는 것이다. 그러면 나중에 XML문서에 있는 보의 폭과 춤 등으로부터 보 단면의 출력에 필요한 VML을 재구성하고 이를 웹에 출력할 수 있다.

3. XML에 기반한 구조계산서의 자료모델

3.1 자료모델의 구성

전자화를 염두에 두고 객체지향설계법(object-oriented design method)(Rumbaugh 등, 1991)에 따라 구조

계산서를 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. 여기에 다시 각 장, 절, 항 등에는 제목(title)이 포함되어 있다는 것을 알 수 있다. 이에 따라 먼저 '구조계산서', '장', '절', '항', '제목' 등의 객체(object)를 도출하고, 이들 사이에 집합관계를 설정할 수 있다. 하지만 이러한 분석 결과는 구조계산서 각 부분의 내용과 형식을 제대로 반영하지 못한 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 3은 이러한 점을 반영하여 정의한 구조계산서 자료모델의 구성을 Unified Modeling Language의 클래스도표(class diagram)로 나타낸 것이다. 그림 3에서 구조계산서를 구성하는 기본적인 단위체는 XShtElement 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 XShtElement 클래스에서 파생되고(derived) 각각에 필요한 자료들이 추가된다. 이에 따라 각 클래스들은 기본적인 단위체로서의 성질을 유지하고 여기에 각각의 다양한 내용과 형식도 적절히 표현할 수 있다.

그림 3에서 'Project', 'Design Information' 등은 정의된 많은 수의 클래스들을 분류하여 간편하게 참조할 수 있는 도표로 표현하기 위한 패키지(package)이다. 여기에서는 현대 건설의 구조계산서 작성지침(현대건설, 1988)을 참고하여 일반적인 철근콘크리트 구조물의 구조계산서에 포함되는 내용을 패키지로 정의하였다.

3.2 자료모델 상세

여기에서는 앞 절에서 기술한 자료모델의 구성에 따라 정의한 자료모델 중 Project와 Design Information 패키지에 대해서만 기술한다. 이 패키지에서는 프로젝트와 설계 개요에 관련된 자료들을 그림 4의 클래스들로 정의한다. 그림 4에서 기본 단위체를 나타내는 XShtElement 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

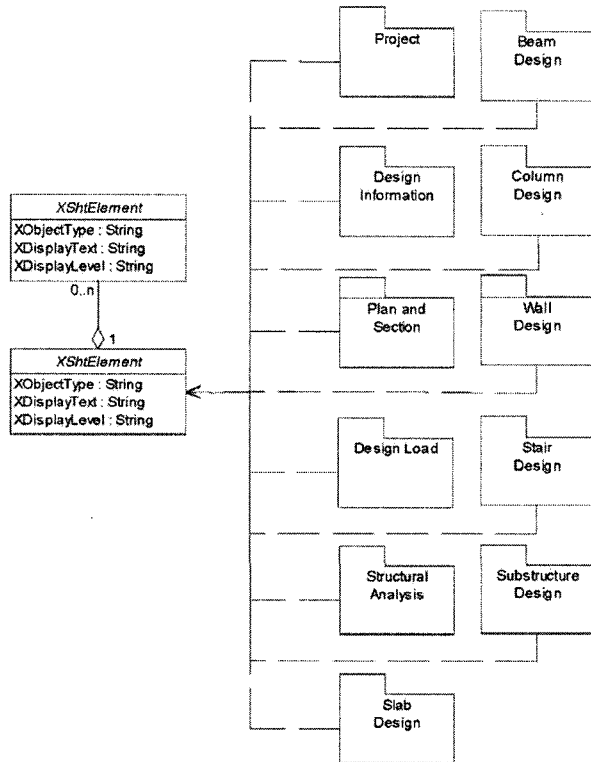


그림 3. 구조계산서 자료모델의 구성

나타내는 XRC, XSteel 클래스가 XMat 클래스에서 파생되므로, 이 클래스들 역시 기본 단위체로 사용된다. 반면에 구조설계의 진행과정과 층고를 나타내는 XProcess와 XStryHgt 클래스는 XShtElement 클래스로부터 파생되지 않았으므로 기본적인 단위체로는 사용될 수 없다. 이 자료모델은 구조계산서를 XML문서로 작성하기 위한 것이므로, 자료모델의 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.

3.3 자료모델의 XML Schema

XML문서는 그 구조, 즉 태그를 이용하여 표현되는 요소(element)와 속성(attribute), 그리고 그 관계를 정의한 Document Type Declaration(이하 DTD)을 따라야 한다. 하지만, DTD는 자료형(data type)이 제한적이고 사용하기 어려운 단점이 있다. 따라서 본 연구에서는 구조계산서를 DTD의 대안으로 개발된 XML Schema를 이용하여 앞서 기술한 자료모델에 따라서 구조계산서 XML Schema를 정의하였다.

그림 5는 그림 4에서 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 종류

는 XBldgInfo 요소의 하위요소로 정의되어 있고, 각종의 층고와 지상으로부터의 높이는 XStryHgt 요소의 하위요소로 정의된 것을 확인할 수 있다.

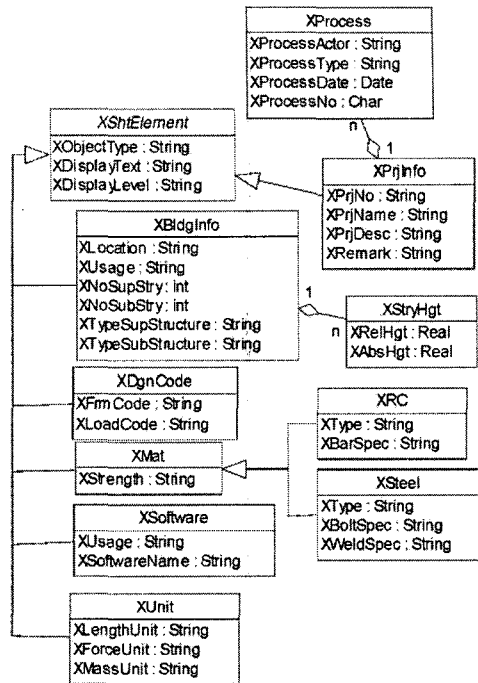


그림 4. Project와 Design Information의 자료모델

```

<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="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 5. XBldgInfo 클래스의 편집된 XML Schema

3.4 자료모델의 데이터베이스 schema

본 연구에서는 구조계산서를 데이터베이스에 저장하기 위해서 앞에서 기술한 자료모델의 데이터베이스 schema를 정의하였다. 데이터베이스로는 Microsoft사의 개발자용 SQL Server 2000을 이용하였다. 그림 6은 Project와 Design Information 패키지의 데이터베이스 schema를 도표로 표현한 것이다.

SQL Server 2000은 관계형 데이터베이스이기 때문에 앞의 자료구조에서 정의된 클래스들을 기본적으로 테이블(table)로 정의한다. 그리고 각 테이블에는 식별자로 XId 필드(field)를 추가한다. 그리고 클래스들 사이의 집합관계는 XParentId 필드를, 일반화관계는 XSuperId 필드를 이용하여 일대다(1:n)관계로 표현한다. 그리고 그림 6에는 나타나지 않았지만 각 자료 사이의 복잡한 관계를 foreign key, constraint, trigger 등을 이용하여 설정하였다.

4. 구조계산서 XML문서의 웹 기반 교환 사례

본 연구에서는 3장에서 제시한 자료모델의 타당성을 검토하기 위해서 지상 20층의 규모의 사무실 건물을 임의로 가정하여 그림 3의 'Project', 'Design Information', 'Plan and Section', 'Design Load', 'Beam Design' 부분만 구조계산서 XML문서로 작성하였다. 구조계산서는 양이 방대하기 때문에 하나의 XML문서로 작성하면 이를 저장, 참조, 편집할 때에 많은 부하가 발생한다. 따라서 이러한 점을 고려하여 본 연구에서는 그림 3에서 패키지로 표현한 부분들을 각각의 장(chapter)으로 설정하고, 이를 단위로 하여 각 장에 해당되는 XML문서를 분리하여 따로 작성하였다.

그림 7은 Beam Design 패키지에 포함되는 자료들의 XML문서이다. 가장 상위에 보 설계하는 부분을 나

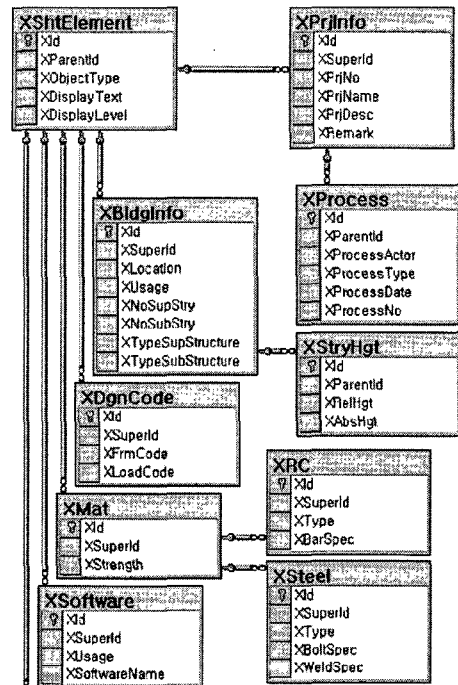


그림 6. Project와 Design Information의 데이터베이스 schema 일부

```

<XDgnSht>
  <XBmDgnSht>
    <XBmDgnShtDisplay> . . . </XBmDgnShtDisplay>
    <XBm>
      <XName>G1</XName>
      <XMatType>Concrete</XMatType>
      <XHgt>500</XHgt>
      <XWth>300</XWth>
      <XStrength>&lt;math>\lt;/math></XStrength>
      <XBarStrength> . . . </XBarStrength>
      <XCover>60</XCover>
      <XBmSect>
        <XSectName> . . . </XSectName>
        <XLoc>Left</XLoc>
        <XStirrupSize>10</XStirrupSize>
      </XBmSect>
    </XBm>
  </XBmDgnSht>
</XDgnSht>

<XStirrupSp>210</XStirrupSp>
<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 7. Beam Design 패키지의 편집된 XML문서

타내는 <XBmDgnSht>가 있다. 여기에는 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. 그림 7에서 밑줄이 그어진 <XStrength> 요소는 하중의 크기를 나타내는데, 그 값이 $f_{ck} = 240\text{kgf/cm}^2$ 에 해당되는 MathML인 '$...$'로 되어 있다. 여기에서 MathML에 포함되는 태그의 시작과 끝을 나타내는 '<'와 '>'는 XML문서에서 보통 문자로는 사용할 수 없는 'reserved keyword'이므로 '<'와 '>'를 의미하는 '<'와 '>'로 표현하였다.

본 연구에서는 XML로 표현된 구조계산서를 웹에서 교환할 수 있는 간단한 웹 응용프로그램을 XHTML, DOM, JavaScript를 이용하여 개발하였다. 그림 8은 구조계산서 중에서 보의 설계 결과를 나타낸 XML문서를 웹을 통하여 교환하고 출력한 화면으로서, 철근콘크리트 보의 왼쪽, 중앙부 단면의 배근 상세가 VML을 이용한 그래픽으로 나타나며 상부에는 콘크리트의 강도가 MathML을 이용한 수식으로 나타난다.

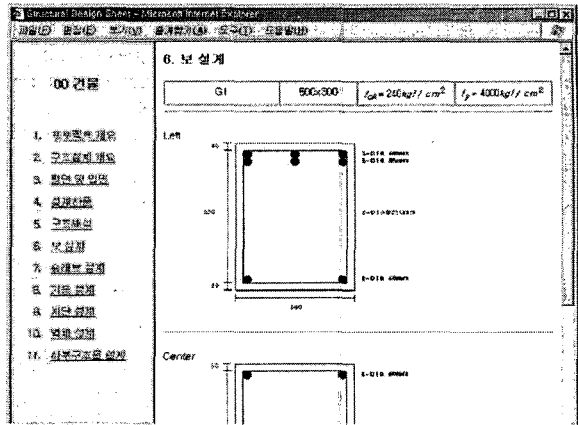


그림 8. Beam Design 출력 화면의 일부분

5. 결론

본 연구에서는 구조계산서를 XML로 표현하기 위한 자료모델을 정의하였으며, 이상의 연구과정을 통하여 다음의 결론을 얻었다.

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

둘째, 구조계산서의 수식은 MathML을 이용하되 XML문서에 MathML을 직접 추가하는 방법이 효과적이다. 그래픽의 경우에는 VML을 이용하되, XML문서에 VML을 직접 추가하기 보다는 VML의 재구성에 필요한 자료들만 포함시키는 것이 효과적이다.

감사의 글

이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2004-003-D00439)

참고문헌

- 현대건설 (1988) 설계실무지침 - 구조계산서 작성지침, 현대건설 해외건축사업본부 건축연구실 발생도서 58, HDM F-0002, p.279
- James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorenson (1991) *Object-Oriented Modeling and Design*, Prentice Hall, p.500

다. 그래픽의 경우에는 VML을 이용하되, XML문서에 VML을 직접 추가하기 보다는 VML의 재구성에 필요한 자료들만 포함시키는 것이 효과적이다.

감사의 글

이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2004-003-D00439)

참고문헌

- 현대건설 (1988) 설계실무지침 - 구조계산서 작성지침, 현대건설 해외건축사업본부 건축연구실 발생도서 58, HDM F-0002, p.279
- James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorenson (1991) *Object-Oriented Modeling and Design*, Prentice Hall, p.500