

데이터베이스와 XML에 기반한 건설프로젝트 기술문서 전자화

Digitalizing Technical Documents of Construction Projects Based on Database and XML

정 종 현*

Jung, Jong-Hyun

요 약

본 연구의 목적은 건설프로젝트에 관련된 기술문서(technical document)를 효율적으로 저장하고 웹(web)을 통하여 신속하고 정확하게 교환할 수 있도록 전자화(digitalization)하는 방안을 제시하는 것이다. 이를 위해 본 연구에서는 먼저 전자화 측면을 고려하여 기술문서의 여러 특성을 분석하였고 이를 바탕으로 저장에는 데이터베이스(database)를, 웹을 통한 교환에는 XML을 적용하여 기술문서를 전자화할 수 있는 방안을 도출하였다. 여기에는 기술문서의 저장과 검색, 부분적인 XML문서의 작성, 기술문서에 적합한 구성항목과 구성체계 설정, 웹에서의 수식과 그래픽 처리 방안이 포함된다. 마지막에는 전형적인 기술문서 중의 하나인 구조계산서(structural design sheets)를 대상으로 하여 데이터베이스에 저장하기 위한 자료구조(data structure) 정의, 실제 XML문서의 작성 및 웹을 통한 교환 등을 부분적으로 구현하여 본 연구에서 제안한 데이터베이스와 XML에 기반한 기술문서 전자화 방안의 적정성을 검토하였다.

키워드 : 기술문서, 전자화, 데이터베이스, XML

1. 서 론

1.1 연구의 배경 및 목적

건설프로젝트 수행 과정에서 자료를 교환하는 가장 중요한 수단은 도면(drawing)이다. 이에 따라 도면을 전자화하여 파일(file)로 저장하고 이를 교환하기 위한 연구들이 이루어졌다(한국건설기술연구원 1999a, 2002). 그리고 최근 웹(web)을 통한 자료 교환의 필요성이 증대되면서 도면 혹은 도면에 포함되어 있는 자료들을 웹을 통하여 교환하기 위한 연구들이 수행되었다(이주영 외 2003, 김성아 외 2004).

하지만 건설프로젝트에서는 도면이 아닌 다른 다양한 문서들도 함께 자료의 교환에 이용한다. 따라서 도면의 전자화 성과를 바탕으로 건설프로젝트의 생산성을 향상시키기 위해서는 도면 외에 다른 관련 문서들도 함께 전자화하여 체계적이고 효율적으로 저장하고 교환할 수 있어야 한다. 이러한 필요에 따라 건설프로젝트 관련 문서들을 전자화하기 위한 연구들이 수행되었다(한국건설기술연구원 1999b, 2001, 2003, 정성운 2003). 그리고

이 연구들의 결과물로서 건설CALS/EC 전자문서 표준(V1.0)(한국건설기술연구원 외 2004)이 공표되었다.

건설CALS/EC 전자문서 표준(V1.0)에서는 최근 웹상의 자료 교환 형식(format)으로서 주목받고 있는 eXtensible Markup Language(이하 XML)를 이용하였다. 즉, 건설프로젝트와 관련된 되는 문서들을 각각에 해당하는 Data Type Definition(이하 DTD), XML문서, eXtensible Stylesheet Language(이하 XSL)를 작성하고 여기에 검색을 위한 색인자료(index data)를 추가하여 파일형태로 저장하는 전자화 방안을 선택하였다. 여기에서 DTD는 XML 문서의 자료구조를 정의하고, XSL은 XML 문서를 HTML로 변환하여 웹에 출력한다. 건설프로젝트의 문서들은 크게 '서식문서'와 '기술문서(technical document)'로 구분할 수 있다. 서식문서는 자재소요계획서, 기성고조서, 착공인허가 서류 등과 같이 건설프로젝트의 현황이나 계획 등을 간략히 요약하여 정리한 일종의 행정문서로서 "일정한 형식을 갖" (정성운 2003)고 양이 적으며 주로 문자(text)와 표(table)로 구성된다. 이러한 서식문서의 특성을 고려하면, 건설CALS/EC 전자문서 표준(V1.0)에서 선택한 방안은 서식문서에 적합한 전자화 방안이다.

반면에 기술문서는 "하나의 시설물을 설계하고 시공하는데 필요한 토목, 지반, 역학, 설비, 환경 등 여러 공학기술이나 방법,

* 일반회원, 경남대학교 건축학부 전임강사, 공학박사
이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2004-003-D00439)

규격”(정성운 2003) 또는 시설물을 설계 및 시공하는 데에 필요한 여러 전문분야의 복잡한 공학적 논리(engineering logic)에 따라 수행한 작업의 결과를 체계적으로 정리하여 기록한 문서로서 서식문서에 첨부되는 경우가 많다. 여기에는 각종 시방서, 공정표, 구조계산서 등이 포함된다. 기술문서는 서식문서와는 그 목적과 내용이 다르기 때문에 다른 특성들을 갖는다. 이로 인해 앞에서 기술한 DTD, XML문서, XSL을 색인자료와 함께 파일 형태로 직접 저장하는 전자화 방안을 기술문서에 그대로 적용하기에는 곤란한 점이 있다.

이에 따라 본 연구에서는 기술문서의 여러 가지 특성들을 충분히 반영하고, 또한 웹에 기반한 자료교환이라는 최근 경향을 고려하여 체계적이고 효과적으로 건설프로젝트 기술문서를 전자화(digitalization)할 수 있는 방안을 제안하고자 한다.

1.2 연구의 내용 및 방법

본 연구의 목적을 달성하기 위해서 먼저 건설프로젝트 기술문서들의 여러 특성들을 파악하고 이를 바탕으로 성공적인 전자화에 필요한 요구조건들을 도출한다.

두 번째는 데이터베이스(database)를 자료 저장 도구로 이용하여 복잡하고 양이 방대한 기술문서를 효율적으로 저장 및 검색할 수 있는 방안을 제안한다.

세 번째는 XML을 이용하여 웹을 통해서 서식문서와는 다른 특성을 갖고 있는 기술문서를 신속하고 편리하게 교환할 수 있는 방안을 제안한다.

마지막에는, 전형적인 기술문서인 구조계산서를 대상으로 자료구조를 정의하여 예제 건물에 대한 자료를 데이터베이스에 저장한 후에 이를 XML문서로 작성하고 웹에서 교환함으로써, 본 연구에서 제안한 기술문서 전자화 방안의 적정성을 검토한다.

2. 기술문서 전자화를 위한 요구조건

2.1 기술문서의 특성

기술문서는 서식문서와는 그 목적과 내용이 다르다. 그러므로 기술문서는 전자화와 관련하여 서식문서와는 다른 다음과 같은 특성들을 갖는다.

첫째, 서식문서는, 그 한 예인 착공신고서의 일부를 나타낸 그림 1과 같이, 대부분 그 서식이 정해져 있다. 반면에 기술문서의 서식은 시설물이나 재료의 종류, 규모 등에 따라 달라지며, 또한 기술문서를 작성하는 회사나 작성자에 따라라도 달라질 수 있다. 그러므로 기술문서에 포함되는 내용과 그 상호 관계, 즉 구성항목과 구성체계가 일정하지 않다.

둘째, 서식문서는 자료의 양이 많아도 서너 쪽인 경우가 대부

분이다. 하지만 기술문서는 자료의 양이 적어도 수백 쪽, 많은 경우에는 수천 쪽에 달한다. 그림 2는 기술문서의 전형적인 예로서 구조계산서의 보 설계결과를 나타낸 것인데, 대부분 보가 적어도 수십 개가 되며 각 보마다 단면이 세 개 정도이다. 또한 이 외에도 기둥과 벽체 등에 대해서도 유사한 그림들이 추가된다. 이러한 점을 고려하면 그 양의 방대함을 추측할 수 있다.

[별지 제13호서식] <개정 2003.11.21> (앞쪽)

| | | | |
|-----------|---------------|------------|--------|
| 학 공 신 고 서 | | | |
| 신고인(건축주) | | 주민(법인)등록번호 | |
| 주 소 | (전화번호 :) | | |
| 허가(신고)번호 | | 허가(신고)일 | |
| 대 지 위 치 | | 지 번 | |
| 학 공 예 정 일 | | | |
| 설 계 자 | 성 명 (서명 또는 인) | 면 허 번 호 | |
| | 사무소명 | 등록 번 호 | |
| | 주 소 | (전화번호 :) | 도급금액 원 |

그림 1. 착공신고서의 일부분

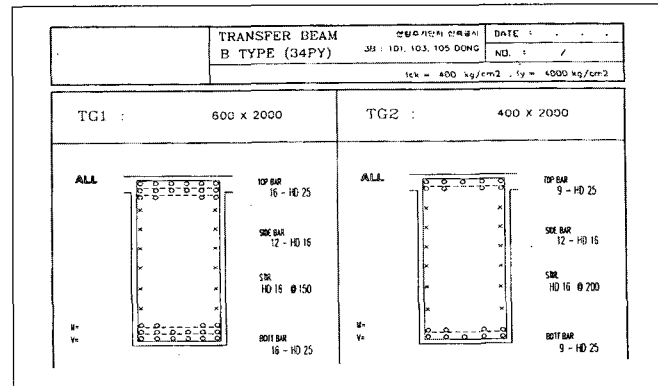


그림 2. 보의 설계결과를 나타내는 구조계산서의 일부분

셋째, 서식문서는 그에 포함되는 자료들의 관계가 비교적 단순하다. 그렇지만 기술문서는 복잡한 공학적 논리에 따라 수행한 작업의 결과이기 때문에 그에 포함되는 자료들이 서로 복잡한 관계를 갖고 있다. 이는 물론 기본적으로 양이 많은 것에서도 비롯된다. 공정표를 예로 들면 각 공정의 개시일과 완료일이 관련된 공정들의 선·후행 관계와 일치해야만 한다.

넷째, 자료들의 표현형식이 다양하다. 서식문서의 경우에는 그림 1에서 확인할 수 있는 것처럼 대부분 문자와 표로 이루어진다. 반면에 기술문서에는 그림 2에서 확인할 수 있는 것처럼 자료들이 문자와 표뿐만 아니라 각종 이미지(image), 그래픽(graphics), 수식(mathematical expressions) 등 다양한 형식으로 표현되어야 한다.

2.2 기술문서 전자화를 위한 요구조건

앞 절에서 기술한 특성들을 반영하여 기술문서를 효과적으로 전자화하기 위해서는 다음과 같은 요구조건들이 만족되어야 한다.

첫째, 기술문서는 양이 방대하기 때문에 효율적인 저장과 신속하고 정확한 검색이 가능해야 한다. 여기의 검색에는 기술문서 내부의 자료를 이용하여 특정 기술문서 전체나 그 일부분을 검색하는 것도 포함된다. 이를 위해서는 자료의 저장과 검색에 큰 장점을 갖고 있는 데이터베이스를 도입하고 XML과 연계하여 적극적으로 활용하는 것이 바람직하다. 데이터베이스를 도입하면 기술문서에 포함되어 있는 자료들 사이에 존재하는 복잡한 관계도 효과적으로 관리할 수 있다.

둘째, 기술문서는 구성체계와 구성항목이 일정하지 않기 때문에 기술문서를 필요에 따라 적합한 체계와 항목으로 구성할 수 있어야 한다. 이를 위해서는 기술문서에 포함되는 자료들을 작은 단위체들로 구분하고 다시 이 단위체들을 적절히 조합하여 필요한 구성체계와 항목을 갖는 XML문서를 작성할 수 있어야 한다.

셋째, 기술문서는 양이 방대하기 때문에 웹을 통하여 기술문서를 교환하는 데에 소요되는 시간과 자원(resource)을 줄여야 한다. 이는 기술문서 전체가 아니라 필요한 부분만을 데이터베이스로부터 추출하여 XML문서로 작성하고 웹을 통해서 교환하는 방식을 통해서 이루어 질 수 있다.

넷째, 기술문서에는 문자 뿐 아니라 표, 이미지, 그래픽, 수식 등 다양한 형태의 자료들이 포함되어 있으므로 이를 효과적으로 처리해야 한다. 특히 데이터베이스와 웹에서는 그래픽과 수식을 직접 처리할 수 없어 그래픽과 수식을 다른 프로그램으로 작성한 후 이미지로 변환하여 처리하고 있다. 기술문서에 포함되어 있는 수식과 그래픽의 양과 복잡함을 고려한다면 수식과 그래픽을 직접 데이터베이스에 저장하고 이를 XML문서로 작성하여 웹에서 처리할 수 있는 방안을 마련해야 한다.

3. 데이터베이스 활용체계

이 장에서는 기술문서를 효율적으로 저장하고 검색하기 위해서 데이터베이스를 도입하여 XML과 연계시키는 방안에 대해서 기술한다.

3.1 저장과 교환의 분리

기술문서의 전자화에 데이터베이스와 XML을 연계하여 활용하는 방법으로는 두 가지가 있다. 첫째는 기술문서를 XML문서로 작성한 후에 XML문서의 내용 전체를 하나의 문자열(string)

로 데이터베이스에 저장하는 것이다. 이 경우에는 검색을 위해서 색인자료도 데이터베이스에 저장한다. 웹을 통해서 교환해야 하는 경우에는 색인자료를 이용하여 해당 문자열을 검색하고 이를 추출하여 그대로 XML문서로 작성한다. 이 방법은 기술문서의 저장과 교환을 분리하여 저장과 검색에는 데이터베이스의 장점을 활용하고 기술문서의 교환에는 XML을 활용하는 방법이다. 하지만 XML문서에 포함되어 있는 자료들을 바탕으로 검색하는 데에 한계가 있으므로 데이터베이스의 장점을 충분히 활용할 수 없다. 또한 필요한 부분만 추출하여 XML문서로 작성할 수 없으므로 기술문서의 저장과 교환에도 많은 시간과 자원이 소요된다. 그리고 기술문서에 포함되는 자료들 사이의 복잡한 관계를 전혀 관리할 수 없다.

둘째는, 본 연구에서 선택한 방법으로, 기술문서에 포함되는 자료들을 추출하여 데이터베이스에 저장하고 필요한 경우에 다시 이 자료들을 조합하여 XML문서를 작성하고 웹을 통해서 교환하는 것이다. 그림 3은 이 방법을 도식화하여 나타낸 것이다. 여기에서 보면 ①번 데이터베이스에 기술문서의 자료들이 나름의 관계에 따라서 저장되어 있다. 그리고 필요에 따라 데이터베이스에 저장되어 있는 해당 자료들만을 추출 및 조합하여 ②번, ③번, ④번과 같이 기술문서의 일부분에 해당되는 XML문서를 작성한다. 이 방법 역시 문서의 저장과 교환을 분리하여 자료의 저장 및 검색과 웹을 통한 교환에 데이터베이스와 XML의 장점을 활용할 수 있는데, 앞의 방법과 달리 기술문서에 포함되어 있는 자료들을 이용하므로 전체 내용에 대한 검색에 제한이 없다. 그리고 데이터베이스에서 제공하는 기능을 이용하여 여러 자료들 사이의 관계를 효과적으로 관리할 수 있으므로 기술문서에 포함되는 자료들 사이에 존재하는 복잡한 관계를 적절히 관리할 수 있는 장점이 있다.

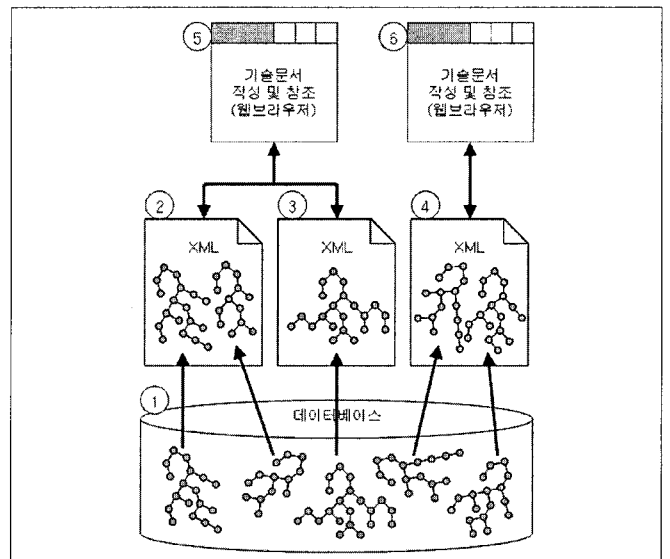


그림 3. 데이터베이스와 XML의 활용체계

기술문서에 포함되는 자료들을 데이터베이스에 저장함에 있어서 주의해야 할 점이 있다. 기술문서는 건설프로젝트에서 활용하는 공학적인 논리와 그 결과를 기록하는 문서이므로 문서로서 갖는 역할과 성질을 적절히 반영해야 한다. 예를 들어 건물의 층고가 3500mm인 경우, 일반적으로는 숫자 3500으로 데이터베이스에 저장해야 하지만, 기술문서를 저장하는 목적이라면 '3500mm' 라는 문자열로 저장하는 것이 바람직하다. 기술문서는 작성과정에서 이미 3500이라는 숫자의 의미를 활용하여 수행한 작업의 결과를 기록한 것이므로 숫자 3500의 의미보다는 기술문서의 일부분을 구성하는 '3500mm' 라는 문자열로서의 의미가 중요하다. 그러므로 기술문서에 포함되는 자료는 일반적인 경우와는 달리 기술문서에서의 역할과 성질에 따라 자료형(data type)과 값을 결정해야 한다.

3.2 기술문서 자료의 분해와 조합

기술문서는 앞에서 이미 기술한대로 그 구성체계와 구성항목이 일정하지 않고, 동일한 종류의 기술문서라 하더라도 시설물의 종류나 작성자에 따라 달라진다. 하지만, 모든 기술문서는 해당되는 내용의 표현에 있어서 위계체계를 이루는 장(chapter), 절(section), 항(subsection) 등으로 구성된다는 공통점이 있다. 그러므로 기술문서의 각 구성항목을 필요에 따라 자유자재로 장, 절, 항 등으로 표현할 수 있도록 하면 각 기술문서를 그에 적절한 구성체계와 구성항목으로 구성할 수 있다.

이에 따라 본 연구에서는 기술문서의 자료를 기본적인 단위체, 즉 구성항목으로 구분하고 각각에 대해 제목, 구성체계 등을 명시적으로 표현하는 자료들을 추가하여 데이터베이스에 저장하도록 한다. 이렇게 되면 필요한 구성항목들을 데이터베이스로부터 추출하여 각 기술문서에 합당한 구성체계에 따라 구성항목을 조합하고 이를 XML문서로 작성할 수 있다.

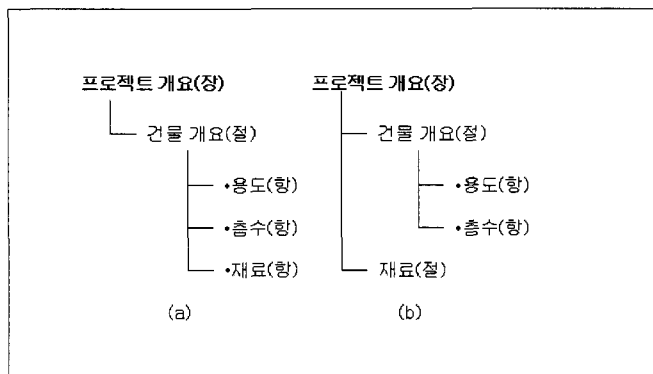


그림 4. 기술문서의 구성항목과 구성체계의 예

그림 4의 예를 보면 구성항목으로 '프로젝트 개요', '건물 개요', '용도', '층수', '재료'가 있는데, (a)와 (b)에서 그 구성체

계가 서로 다르다. 이런 경우에는 표 1에 나타난 바와 같이 각 구성항목을 단위체로 설정하고 (a)와 (b) 각각에 적합한 제목, 구성체계로서의 종류와 포함하는 항목을 추가하여 데이터베이스에 저장한다. 식별자는 각 구성항목을 구별하기 위한 일련번호이다.

표 1. 구성항목과 구성체계의 값의 예

| 식별자 | 제목 | 종류 | | 포함항목 식별자 | |
|-----|---------|-----|-----|----------|------|
| | | (a) | (b) | (a) | (b) |
| 1 | 프로젝트 개요 | 장 | 장 | 2 | 2, 5 |
| 2 | 건물 개요 | 절 | 절 | 3, 4, 5 | 3,4 |
| 3 | 용도 | 항 | 항 | - | - |
| 4 | 층수 | 항 | 항 | - | - |
| 5 | 재료 | 항 | 절 | - | - |

4. XML 활용체계

이 장에서는 XML을 기술문서의 특성에 적합한 방식으로 활용할 수 있는 방안에 대해서 기술한다.

4.1 XML문서의 분해

기술문서는 양이 방대하기 때문에 웹을 통한 교환에 소요되는 시간과 자원을 감소시키기 위해서 기술문서 전체가 아니라 필요한 부분만 교환해야 한다. 즉, 기술문서를 하나의 XML문서가 아니라 여러 개의 XML문서로 나누어 작성하고 교환해야 한다. 이를 위해서 기술문서를 나누어 교환할 부분이 결정되면, 이 부분에 포함되는 자료들을 데이터베이스로부터 검색한다. 다음에는 이 자료들을 조합하여 필요한 부분에 해당되는 XML문서를 작성하고 웹에서 교환한다.

그런데, 이 과정에서 필요하다면 교환할 부분을 다시 보다 작은 여러 부분으로 구분하여 각각에 대한 자료들만으로 XML문서를 작성하고 이를 웹에서 교환할 수도 있다. 예를 들어 기술문서 전체가 아니라 그림 4 (a)의 '건물 개요'만을 웹에서 교환한다면, 우선 '건물 개요', '용도', '층수', '재료' 부분에 포함되는 자료 전체를 데이터베이스에서 추출하고 조합하여 하나의 XML문서로 작성하고 이를 웹에서 교환하는 방법을 고려할 수 있다. 하지만, 만약 '재료'에 해당되는 자료들이 아주 많다면 '건물 개요', '용도', '층수'에 해당되는 자료들을 추출하여 하나의 XML문서로, '재료'에 해당되는 자료들을 하나의 XML문서로 작성하여 이들을 필요에 따라 동시에 혹은 순차적으로 웹에서 교환하는 방법도 고려할 수 있다.

이와 같이 XML문서를 분해하여 교환하는 방식은 그림 3에 표현되어 있다. 그림 3에서 ⑥번은 필요한 부분을 ④번 XML문

서 하나로 작성하고 이를 웹에서 교환한 경우이다. ⑤번은 필요한 부분을 ②번과 ③번의 XML문서로 나누어 작성하여 동시에 웹에서 교환한 경우이다.

4.2 MathML을 이용한 수식의 처리

MathML은 XML을 기반으로 웹이나 일반 소프트웨어에서 수식을 편리하게 처리하는 것을 목적으로 개발되었다. 그림 5는 수식을 MathML문서로 작성한 예이다.

MathML을 적절히 활용하기 위해서는 다음의 두 가지가 해결되어야 한다. 첫째는 MathML로 표현된 수식을 웹에 출력(presentation)하는 방법이다. 그 방법으로는 XML문서를 다른 형태의 문서로 변환할 수 있는 XSL을 이용하는 방법과 특별한 부가적인 처리를 하지 않더라도 웹에서 MathML을 간편하게 출력할 수 있는 plug-in을 이용하는 방법이 있다. 기술문서의 경우에는 그에 포함되는 자료들이 서로 복잡한 관계를 갖고 있기 때문에 XSL을 이용한 출력이 곤란하므로 MathML의 출력에 plug-in을 이용하는 것이 합리적이다. 본 연구에서는 Design Science사의 MathPlayer plug-in (Design Science Homepage, 2005)을 이용한다.

```
<math>
  <msup>
    <mi>b</mi><mn>2</mn>
  </msup>
  <mo>=</mo><mn>3</mn>
</math>
```

그림 5. MathML을 이용한 수식의 표현 예

둘째는 데이터베이스에 기술문서의 수식에 필요한 MathML을 저장할지 아니면 해당 MathML을 재구성할 수 있는 자료를 저장할지의 여부이다. 그림 5의 예를 든다면 그림 5의 MathML 자체를 문자열로 저장할지 아니면 'b', '2', '3' 등의 기본적인 자료들을 저장하고 향후에 'b', '2', '3'을 이용하여 그림 5의 MathML을 다시 재구성할지의 여부이다. 기술문서에 포함되는 수식은 대부분 이미 결정되어 있으므로 이에 해당되는 MathML을 미리 작성하여 데이터베이스에 저장하는 것이 효과적이다. 또한 수식의 의미를 파악하여 다른 작업을 수행할 필요가 없으므로 수식에 필요한 자료로부터 MathML을 재구성하는 것은 불필요한 작업이다.

4.3 VML을 이용한 그래픽의 처리

Vector Markup Language(이하 VML)는 웹에서 벡터(vector) 방식의 그래픽을 이용하기 위한 것으로서, MathML과

마찬가지로 XML을 기반으로 하므로 일반적인 XML문서와 동일하다. 그림 6은 붉은 색의 직사각형을 출력하기 위한 VML문서의 예이다.

VML은 Microsoft사의 Internet Explorer에서 직접 지원하므로 별도의 XSL이나 기타 plug-in 등이 필요 없다. 그러므로 일반적인 XML문서에 원하는 그림에 대한 VML을 추가하면 특별한 처리를 하지 않더라도 그래픽이 출력된다. 따라서 VML을 이용하면 아주 편리하게 웹에서 그래픽을 처리할 수 있다. 하지만, VML은 아직 웹에서 XML을 기반으로 그래픽을 처리하는 표준이 아니어서 모든 웹브라우저에서 지원하지 않는 단점이 있다.

기술문서에 포함되는 그래픽은 종류가 다양할 뿐만 아니라 그 수도 매우 많으므로 그에 해당되는 VML을 모두 미리 작성하여 데이터베이스에 저장하는 것은, 수식의 경우와는 달리, 비효율적이다. 따라서 기술문서의 경우에는 데이터베이스에 그래픽을 표현하는 VML을 문자열로 직접 저장하지 않고 해당 VML을 재구성하기 위해서 필요한 자료들만 저장한다. 예를 들어 사각형 그래픽이 필요하다면 그림 6의 VML을 문자열로 추가하는 것이 아니라 사각형의 가로와 세로의 길이, 색 등을 데이터베이스에 저장하는 것이다. 그러면 나중에 데이터베이스에 있는 사각형의 가로와 세로의 길이, 색 등으로부터 사각형의 출력에 필요한 VML을 재구성하여 웹에 출력할 수 있다.

기술문서에는 복잡한 렌더링(rendering)이 필요 없는 비교적 단순한 2차원 그래픽이 대부분이기 때문에 이 방식이 가능하며, 나중에 VML이 변경되는 경우에 효과적으로 대비할 수 있고 데이터베이스에 저장되는 자료의 양도 상당히 줄어든다.

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

그림 6. VML을 이용한 그래픽의 표현 예

5. 적용 사례

지금까지 데이터베이스와 XML을 이용한 기술문서의 전자화 방안을 기술하였다. 본 장에서는 구조계산서(structural design sheets)와 지상 20층 규모의 사무실 건물을 예로 이 방안의 타당성을 검토한다.

본 연구에서 사례적용의 대상으로 선정한 구조계산서는 구조 해석 및 구조설계를 수행하는 과정에서 생성되는 여러 자료들과 최종 결과물을 정리하여 기록한 문서로서, 그 내용이 복잡하고 분량이 매우 방대하며, 많은 수의 수식과 그래픽이 포함되어 있어 기술문서의 전형적인 특성을 잘 보여준다.

5.1 구조계산서의 개념적 자료구조

본 연구에서는 개념적 자료구조의 정의에 객체지향설계법(object-oriented design method)을 이용하였으며, 여기에서는 지면관계상 구조계산서에 포함되어야 하는 건설프로젝트와 건물에 대한 자료구조만 그림 7에 Unified Modeling Language의 클래스 도표(class diagram)로 표현하였다.

구조계산서는 여러 개의 장으로 구성되는데, 각 장은 여러 절을, 다시 각 절은 여러 항을 포함한다. 물론 각 항은 다시 더 구체적으로 분해될 수 있다. 그리고 각 장, 절, 항에는 제목(title)이 포함되어 있다. 그러므로 구조계산서는 위계적인 집합관계(aggregation)를 갖는 여러 단위체로 구성된다. 이를 반영하여 본 연구에서는 3.2절에서 기술한 대로 기술문서의 구성체계와 구성항목을 원하는 대로 구성하기 위한 기본적인 단위체로서 XShtElement 클래스(class)를 정의한다. 그림 7에 나타난 대로 이 클래스에는 3개의 속성(attribute)이 포함되는데, XDisplayText는 제목을, XDisplayLevel은 해당 단위체의 유형, 즉 장, 절, 항 등을 표현한다. 그리고 자기 자신과 일대다(1:n)의 집합관계를 갖는다. 기술문서의 기본적인 단위체로서의 역할을 하는 프로젝트 개요, 건물 개요, 설계기준, 재료, 사용 소

프트웨어, 단위를 나타내는 XPrjInfo, XBldgInfo, XDgnCode, XMat, XSoftware, XUnit 등의 클래스들은 XShtElement 클래스로부터 파생되어 그에 포함되어 있는 속성들을 모두 상속받아 기본적인 단위체로 사용될 수 있다. 이는 XMat 클래스로부터 파생되어 RC와 철골을 나타내는 XRC, XSteel 클래스의 경우에도 마찬가지 이다. 반면에 구조설계의 진행과정과 층고를 나타내는 XProcess, XStryHgt 클래스는 XShtElement 클래스로부터 파생되지 않았기 때문에 기본적인 단위체로서는 사용될 수 없다.

5.2 구조계산서의 물리적 자료구조

본 연구에서는 데이터베이스로 Microsoft사의 개발자용 SQL Server 2000을 이용하였으며 5.1절에서 기술한 개념적 자료구조를 물리적 자료구조로 정의하였다. 그림 8은 그림 7에 해당되는 물리적 자료구조를 간단한 도표로 표현한 것이다.

그림 7. 프로젝트 및 건물에 대한 물리적 자료구조SQL Server 2000은 관계형 데이터베이스이기 때문에 개념적 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 8에는 나타나지 않았지만 각 자료 사이의 복잡한 관계를 foreign key, constraint, trigger 등을 이용하여 설정하였다.

데이터베이스의 물리적 자료구조를 정의한 후에는 예제건물에 대한 가상의 구조해석 및 구조설계 결과를 데이터베이스에 입력하여 저장하였다. 이 입력과정에서 자료형이 틀리거나 자료들 사이의 관계에 부합되지 않는 자료를 입력하는 오류가 여러 차례 있었는데, 그때마다 데이터베이스에서 자료의 입력이 자동으로 차단되어 입력오류를 수정할 수 있었다. 이를 통해서 데이터베이스를 도입함으로써 기술문서에 포함되어 있는 자료들 사이의 복잡한 관계를 효과적으로 관리할 수 있음을 확인하였다.

이러한 과정을 거쳐서 예제건물에 대한 가상의 구조계산서 입력을 완료하였으며, 이를 통하여 그림 8의 자료구조를 이용하여 구조계산서에 포함되는 자료들을 분해하고 데이터베이스에 체계적으로 저장할 수 있으며, 또한 5.1절에서 기술한 대로 기본적인 단위체인 XShtElement 클래스를 이용하여 구조계산서의 구성항목과 구성체계를 필요에 따라 설정할 수 있다는 것을 확인하였다. 또한 자료의 저장에 필요한 기억장소(memory)의 양이 상당히 감소하여 구조계산서의 방대한 자료들을 효과적으로 데이터베이스에 저장할 수 있다는 가능성도 충분히 확인하였다. 보의 구조설계 결과를 예로 들면 보 1개당 XML문서의 경우 평

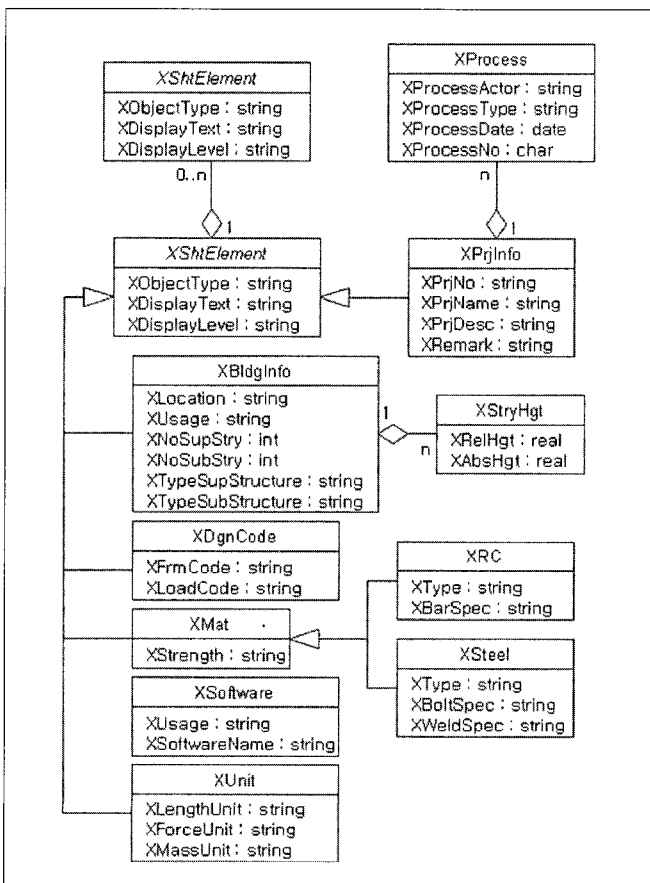


그림 7. 프로젝트 및 건물에 대한 개념적 자료구

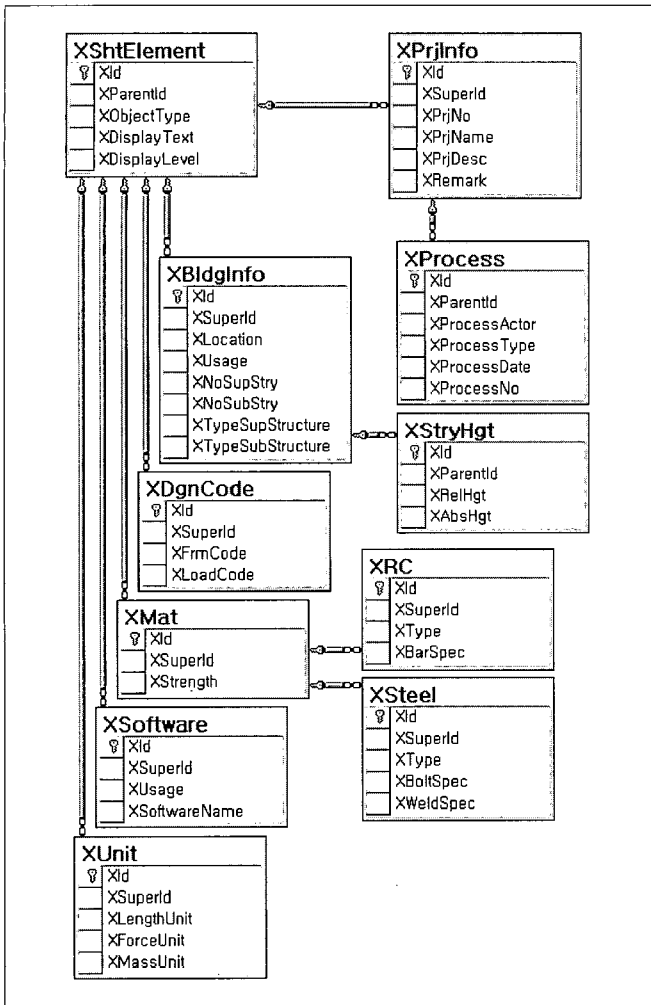


그림 8. 프로젝트 및 건물에 대한 물리적 자료구조

군 4.3KB였지만 데이터베이스에 저장하는 경우 최대 2.8KB, 평균 1.4KB였다. 이 결과로 보면 XML문서를 데이터베이스에 저장하면 기억장치가 최소 35%, 평균 67%정도 감소한다.

5.3 구조계산서의 XML문서

여기에서는 구조계산서를 ‘프로젝트 개요’, ‘구조설계 개요’ 부터 ‘벽체설계’, ‘하부구조물 설계’ 까지 11개의 장으로 구성하였으며, 이 각각의 장을 참조 단위로 설정하여 한 번에 하나의 장에 해당되는 자료들만을 데이터베이스에서 추출하여 XML문서로 작성하고 웹에서 교환하도록 하였다.

데이터베이스의 자료를 추출하여 XML문서로 작성하는 방법은 여러 가지가 있는데, 여기에서는 XML Schema로 XML문서와 데이터베이스를 연결시키는(mapping) 방법을 이용하였다. 이를 위하여 각 장에 해당되는 XML문서의 자료구조를 XML Schema로 정의하였다.

그림 9는 5.2절에서 기술한 물리적 자료구조에 따라 실제 자료를 입력하고 이 중에서 ‘구조설계 개요’ 장에 해당되는 부

분의 자료만을 데이터베이스로부터 추출하여 XML문서로 작성한 것을 편집하여 간략하게 나타낸 것이다. 그림 9에서 하부에 밑줄로 표시된 (XStrength) 요소는 철근콘크리트의 강도를 나타내는데 그 값은 ‘240 <math>... math>’로 표현되는 MathML이다. 재료의 강도는 정수(integer)나 실수(real)이지만, 구조계산서에 단위와 함께 나타낼 수 있도록, ‘kgf/cm²’에 해당되는 MathML을 추가하여 문자열(string)로 데이터베이스에 저장하였고 이에 따라서 XML문서에도 해당 MathML이 나타난다. 그리고 MathML에 포함되는 태그(tag)의 시작과 끝을 나타내는 ‘<’와 ‘>’는 XML 문서에서 보통 문자로는 사용할 수 없는 ‘reserved keyword’이므로 ‘<’와 ‘>’를 의미하는 ‘<’와 ‘>’로 표현하였다.

5.4 구조계산서 XML문서의 교환

그림 10과 11은 구조계산서를 웹에서 교환한 결과화면이다.

```
<?xml version="1.0" standalone="yes"?>
<XDgnInfo XId="2">
  <XObjectType>XDgnInfo</XObjectType>
  <XDisplayText>2. 구조설계 개요</XDisplayText>
  <XDisplayLevel>Chapter</XDisplayLevel>
  <XBldgInfo XId="1">
    <XLocation>서울</XLocation>
    <XUsage>사무실</XUsage>
    <XNoSupStry>20</XNoSupStry>
    ...
  <XBldgInfoTitle XId="3">
    ...
  <XDisplayLevel>Section</XDisplayLevel>
  </XBldgInfoTitle>
  <XStryHgt>
    <XId>01F</XId>
    <XRelHgt>500</XRelHgt>
    ...
  </XStryHgt>
</XBldgInfo>
...
<XMatList XId="5">
  ...
  <XDisplayLevel>Section</XDisplayLevel>
  <XRC XId="1">
    <XType>일반콘크리트</XType>
    <XBarSpec>SD4000</XBarSpec>
    <XRCMat>
      <XStrength>240 &lt;math>... math&gt;</XStrength>
    </XRCMat>
  </XRC>
  ...
</XMatList>
...
</XDgnInfo>
```

그림 9. 프로젝트 및 건물에 대한 XML 문서

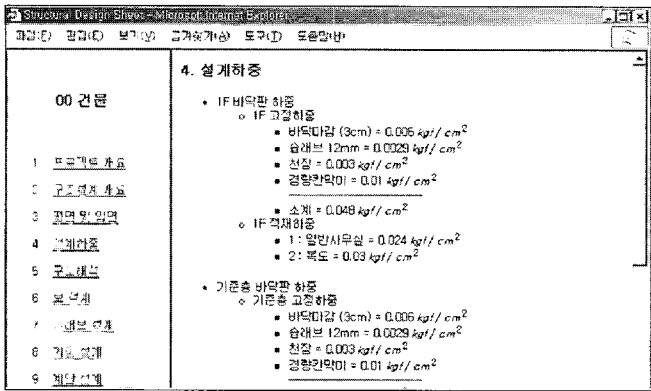


그림 10. 설계하중 출력 화면

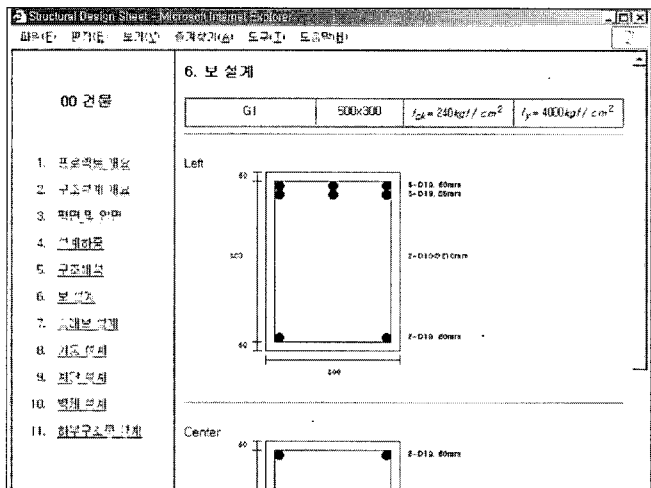


그림 11. 보 설계 출력 화면

여기에서 왼쪽은 구조계산서의 목차이다. 이 목차는 데이터베이스에서 기술문서의 기본 단위체를 나타내는 XShElement 테이블의 XDisplayLevel 값이 '장' 인 것들만 검색하여 XML문서로 작성한 것이다. 목차에는 하이퍼텍스트(hypertext) 기능을 적용하였으므로, 원하는 장을 선택하면 그에 해당되는 부분을 데이터베이스로부터 추출하여 XML문서로 작성하고 오른쪽에 나타나도록 하였다. 이상의 목차와 선택한 부분을 XML문서로 작성하여 교환한 결과를 통해서 기술문서에 포함되어 있는 자료들을 이용하여 기술문서를 신속하고 정확하게 검색할 수 있다는 것을 확인하였다. 그리고 이 적용사례처럼 필요한 부분만 XML문서로 작성하여 교환하면 구조계산서 전체를 한꺼번에 교환하는 것보다 시간과 자원을 절감할 수 있다는 것은 확실하다.

그림 10은 구조계산서 중에서 설계하중을 표현한 것인데, 지붕층이나 기준층 등의 평면 유형별로 그에 해당되는 고정하중과 적재하중이 그 구체적인 내역과 함께 문자와 수식으로 표현된다. 이를 통해서 구조계산서에 포함되어 있는 수식은 해당 MathML을 문자열로 데이터베이스에 저장하고 XML문서에 포함시켜 웹에서 교환하는 방안을 이용하여 적절히 처리할 수 있

음을 확인할 수 있다.

그림 11은 보의 설계 결과를 나타낸 것으로, 철근콘크리트 보의 왼쪽, 중앙부 단면의 배근 상세가 그래픽으로 나타난다. 그림 11에서는 그래픽에 필요한 자료들을 데이터베이스에 저장하고 이를 VML로 재구성하여 웹에서 출력하는 방안이 구조계산서의 그래픽 처리에 적합하다는 것을 확인할 수 있다.

6. 결론

본 연구에서는 데이터베이스와 XML을 이용하여 건설프로젝트 기술문서에 대한 효과적인 전자화 방안을 제안하였고, 사례 적용을 통하여 그 타당성을 검토하였다. 이상의 과정을 통하여 다음의 결론을 얻었다.

첫째, 기술문서에 포함되는 자료들을 데이터베이스에 저장하고, 이 자료들을 조합하여 XML문서를 작성하고 웹에서 교환하는 방식이 건설프로젝트 기술문서의 효율적인 저장 및 검색과 웹을 통한 신속하고 정확한 교환을 위한 전자화에 합리적인 방안이다. 또한 이러한 방식을 통해서 기술문서에 포함되는 자료들 사이에 존재하는 복잡한 관계를 효율적으로 관리할 수 있다.

둘째, 기술문서를 여러 기본적인 단위체로 분해하여 저장하고 필요한 경우에 해당 단위체들을 적절히 추출 및 조합하여 XML문서로 작성하는 방식이 구성체계와 구성항목이 일정하지 않은 기술문서의 전자화에 적합하다.

셋째, 기술문서 전체가 아니라 필요한 부분만 데이터베이스로부터 추출하여 XML문서로 작성하고 웹을 통해서 교환하는 방식이 양이 방대한 기술문서의 교환에 적합하다.

넷째, 수식은 MathML을 이용하되 수식에 해당되는 MathML을 문자열로 직접 처리하는 방법이 효과적이다. 그래픽은 VML을 이용하면 웹에 간편하게 출력할 수 있으며, 데이터베이스에 해당 VML을 문자열로 직접 저장하기 보다는 VML을 재구성하기 위해서 필요한 자료들만 저장하는 것이 바람직하다.

다섯째, 구조계산서를 대상으로 일부 구현해 본 결과, 데이터베이스와 XML을 이용한 저장 및 웹기반의 교환이 무리 없이 이루어져, 본 연구에서 제시한 방안을 적용하면 건설프로젝트 기술문서를 효과적으로 전자화할 수 있는 가능성이 충분하다는 것을 확인하였다.

참고문헌

1. Design Science Homepage, <http://www.mathtype.com>, 2005
2. 김성아 외 3인, KOSDIC 포맷의 응용수준별 활용모델 개발,

- 대한건축학회논문집(계획계), 제20권, 제4호, 2004, pp.57-64
3. 이주영, 김인한, XML기반의 건설 분야 설계정보의 표현 및 활용에 관한 연구, 대한건축학회논문집(계획계), 제19권, 제6호, 2003, pp.3-10
 4. 정성윤, 건설분야에서의 XML 기반의 전자 기술문서 개발 방안, 한국건설관리학회 학술발표대회 논문집, 제4회, 2003, pp.573-576
 5. 한국건설기술연구원, 도면정보 표준화에 관한 연구, 한국건설기술연구원, 1999a
 6. 한국건설기술연구원, 전자문서 교환을 위한 응용기술 연구, 한국건설기술연구원, 1999b
 7. 한국건설기술연구원, 건설분야의 전자문서 체계화 방안 연구, 한국건설기술연구원, 2001
 8. 한국건설기술연구원, 건설 분야 도면정보 교환체계 구축방안 연구, 한국건설기술연구원, 2002
 9. 한국건설기술연구원, 건설CALS/EC 표준화 개발(I), 한국건설기술연구원, 2003
 10. 한국건설기술연구원 외, 건설CALS/EC 전자문서 표준(V1.0), KCCS-0002-2004, 2004

논문제출일: 2005.03.31

심사완료일: 2005.08.26

Abstract

This study describes the digitalization of technical documents of construction projects using database for storage and XML for exchange format on the web. First, for this purpose, the requirements for effective digitalization are identified. Second, the strategies for using database and XML are presented. These strategies include the way to store and search for the technical documents, to draw up the XML document for some parts of the technical documents, to arrange the components in their proper hierarchy, to manage the graphics and mathematical expressions in database and XML documents. Finally we discussed the validities of the results of this study by partial implementation for structural design sheets which has all the characteristics of technical documents.

Keywords : Technical documents, Digitalization, Database, XML