

主題

# XML기반의 전자상거래 문서양식 관리

아주대학교 김 형 도

차 례

- I. 서 론
- II. XML(Extensibel Markup Language)
- III. XFDL(Extensible Forms Description Language)
- IV. XForms
- V. 결 론

## 요 약

HTML로 표현된 양식은 데이터 모델을 포함하고 있지 않기 때문에 종이 양식과 같은 전자상거래 레코드를 구성하기 어렵다. XML로 양식을 표현하기 위한 노력은 XFDL이 대표적이다. 전자상거래 언어로서 XFDL의 가치는 데이터, 표현, 계산규칙, 의미등에 대한 정보를 단일한 XML 문서에 통합하는 능력에 있다. XFDL은 고품질의 배치(Layout), 문서화(Documentation) 지원, 통합 계산, 입력 검증, 복수의 중첩되는 디지털 서명, 감사가 가능한 거래 레코드 등을 지원한다. 그러나, 이러한 특징을 갖는 XFDL은 내용과 표현을 분리하고자 하는 W3C의 정책과는 피리가 있다. XForms는 HTML 양식의 한계점을 개선하기 위해서 W3C에서 활발히 추진되고 있는 XML기반의 차세대 양식이다. 핵심 아이디어는 사용자 인터페이스와 표현을 데이터 모델과 논리로부터 분리시켜서, 하나의 폼을 이동단말, 텔레비전, PC등 다양한 용

도로 사용할 수 있도록 지원한다는 점이다. 이 논문에서는 차세대의 양식 표준으로 예상되는 XForms를 중심으로 기업간의 양식 처리를 위한 방안에 대하여 논한다.

## I. 서 론

기업간 거래에서 전달되는 문서를 전자적으로 표현하게 되면, 문서처리에 소요되는 시간과 돈을 절약할 수 있으며, 전달되는 내용의 정확성도 증대시킬 수 있다. 문서에 포함된 데이터의 표현 방법이 양식(Form)으로 정의되듯이, 전자문서에서도 데이터의 표현 방법을 정의할 수 있다. 하나의 문서 데이터에 대해서 다양한 양식이 존재할 수 있으므로 데이터와 양식은 독립적으로 관리하는 것이 바람직하다.

웹상에서 양식은 일반적으로 HTML의 Form 엘리먼트를 사용하여 표현한다. 이 엘리먼트는 폼

컨트롤이라고 부르는 특별한 엘리먼트들과 일반 마크업 엘리먼트들을 포함한다. 사용자는 브라우저상에 표현된 컨트롤에 입력을 제공하여 폼을 완성하고, 입력된 데이터를 서버측에 전달하여 폼과 상호작용한다. 전달되는 폼 데이터는 컨트롤의 이름과 그 값의 쌍으로 이루어진다. 일반적으로 양식은 질문(양식 템플릿)과 대답(입력 데이터)으로 구성되는데, HTML은 대답만을 서버측에 전달한다. 질문이 없이 대답만을 상대방에게 전달하는 것은 특정한 전달 대상만이 그 의미를 파악할 수 있음을 가정하는 것이다. 또한, 질문의 의미가 변화되는 경우 전달되는 내용을 해석하는 에이전트의 구현도 갱신되어야 한다. 양식에 포함된 데이터의 정확한 의미를 전달할 수 없는 HTML 양식은 신뢰성있는 감사 자료가 되기 어렵다.

문서 양식에 관한 정보를 XML로 표현하기 위해서 제안된 XFDL (Extensible Forms Description Language) [1][2][4]은 안전하고 법적인 구속력을 갖는 개방형 프로토콜이다. 전자상거래 언어로서 XFDL의 가치는 데이터, 표현, 계산규칙, 의미등에 대한 정보를 단일한 XML 문서에 통합하는 능력에 있다 [2]. XFDL은 고품질의 배치(Layout), 문서화(Documentation) 지원, 통합 계산, 입력 검증, 복수의 중첩되는 디지털 서명, 감사가 가능한 거래 레코드 등을 지원한다. 그러나, 이러한 특징을 갖는 XFDL은 내용과 표현을 분리하고자 하는 W3C의 정책과는 괴리가 있다.

XForms [8]는 HTML 양식의 한계점을 개선하기 위해서 W3C에서 활발히 추진되고 있는 XML기반의 차세대 양식이다. 핵심 아이디어는 사용자 인터페이스와 표현을 데이터 모델과 논리로부터 분리시켜서, 하나의 폼을 이동단말, 텔레비전, PC등 다양한 용도로 사용할 수 있도록 지원한다는 것이다. 이러한 요구사항을 만족시키기 위해서 XForms는 다른 표준들과의 결합을 고려하여 설계되고 있다. 예를 들면, 표현을 위한 XHTML [6]

과의 결합이나 멀티미디어를 처리하기 위한 SMIL[3]과의 결합 등이 있다.

이 논문에서는 차세대의 양식 표준으로 예상되는 XForms를 중심으로 기업간의 양식 처리를 위한 방안에 대하여 논한다. 먼저 II장에서는 XML에 대하여 살펴보고, III장에서는 XFDL을 사례와 함께 소개한다. IV장에서는 표준화가 진행중인 XForms에 대하여 상세히 검토하고, 마지막으로 V장에서는 XML기반의 양식에 관한 발전 방향을 모색해 본다.

## II. XML(Extensible Markup Language)

웹에서는 멀티미디어 저작물을 손쉽게 생성하고, 인터넷을 통해 서비스를 제공할 수 있도록 HTML을 사용한다. 웹의 폭발적인 보급은 이러한 편리성에서 출발한다고 볼 수 있으며, 앞으로도 각종 저작물이나 문서등 정보의 일방적인 제공에서 매우 중요한 수단이 될 것으로 예측된다. 웹의 상업적인 이용의 활성화와 더불어, 상거래 데이터나 문서의 교환은 매우 중요한 의미를 갖게 된다. 예를 들면, 상품 데이터를 사용자가 수신하여 가격등을 비교하고 최적의 상품을 구매하는 응용들이 여기에 해당한다. 그러나, HTML은 기본적으로 저작물을 화면에 표현하는 방법을 정의하기 위한 마크업 언어로서 데이터의 공유에는 부적절하다. [그림1]과 같이, 상품정보를 테이블 형태로 출력할 경우, <table>...</table>로 그 형태를 정의하고, <tr>...</tr>로 줄을 구분하여 한 제품정보를 구성하며, <td>...</td>로 제품명, 가격등의 정보를 열로 구분하여 표현한다. 이러한 정보는 사람이 분석하여 의미를 파악할 수 있으나, 태그들을 바탕으로 컴퓨터가 그 의미를 해석하는 것은 불가능하다.

이러한 HTML의 문제점을 해결하기 위한 가장 간단한 방법은, 특정한 HTML 페이지의 테이블을

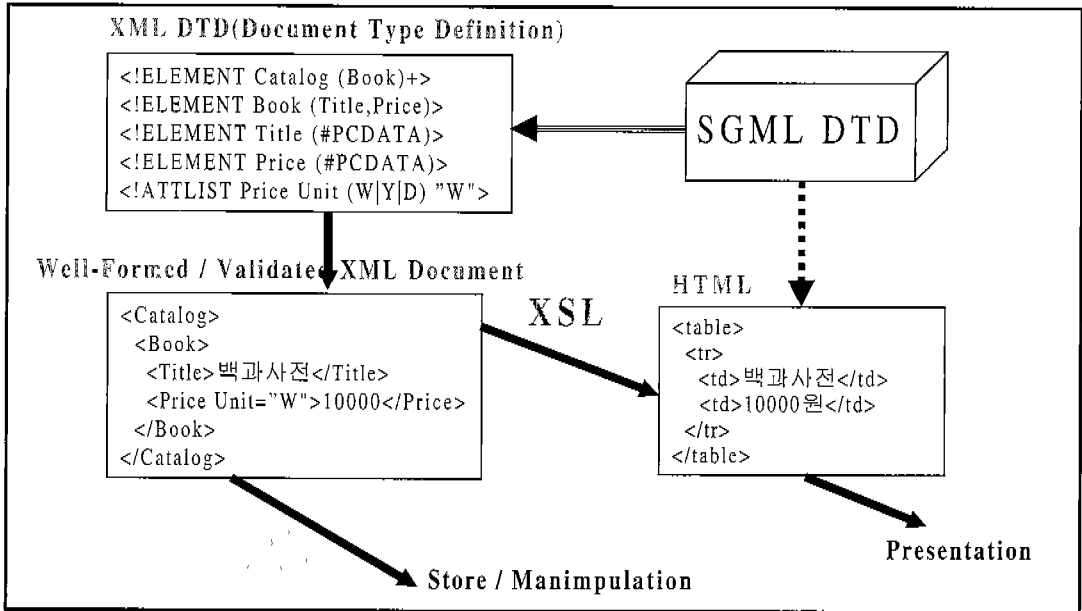


그림 1. XML을 활용한 의미의 전달

구성하는 열 등을 특별하게 해석하는 것이다. 상품 정보 검색 엔진들이 이러한 방법을 사용하고 있으나, 각 페이지별 특성을 추출해서 프로그램으로 작성해야 하며, 구조가 변화된 페이지들을 검색하여 프로그램을 수정해야 하는 등 변화에 취약하다. 또 다른 방법으로는 비표준 태그들을 사용하여 상품정

보를 제공하는 것이다. 이러한 비표준 태그들은 브라우저에서 무시되며, 스크립트와 같은 프로그램에 의해서 특별히 처리된다. 그러나 이런 방법은 여전히 문제가 있다. 즉 여러 사이트에서 다른 형식의 비표준 데이터를 받게 될 경우 각각을 따로 처리해야 하는 문제가 생기게 된다.

표 1. XML, SGML, HTML 비교

| 기준     | XML                                   | SGML                 | HTML                 |
|--------|---------------------------------------|----------------------|----------------------|
| 상호관계   | SGML을 단순화한 것으로 사례정의 가능                | 최초의 마크업 표준으로 사례정의 가능 | SGML로 정의된 한 사례       |
| 용도     | 전자상거래                                 | 문서화                  | 프리젠테이션 용             |
| 확장성    | 가능 (DTD정의 가능)                         | 가능 (DTD정의 가능)        | 불가능 (비표준 확장은 가능)     |
| 복잡도    | 다소 복잡                                 | 매우 복잡                | 간단                   |
| 브라우저   | 현재 IE5.0 지원                           | 전용                   | 범용                   |
| 표준화 현황 | XML1.0권고안 1998년2월<br>XSL, XLL 표준화 진행중 | 1986년 ISO8879 표준     | 1998년HTML4.0권고안      |
| 데이터 공유 | 가능                                    | 가능                   | 불가능                  |
| 스타일 처리 | XSL                                   | DSSSL                | CSS                  |
| 전망     | 웹 변화의 기본 틀<br>표준화 주도<br>기업 경쟁력의 기준    | CALS문서화              | XML의 한 사례로서 정의된 HTML |

HTML의 한계점을 근본적으로 해결하기 위한 방안으로, XML을 사용하는 기술이 속속 실현되고 있다. XML은, 간단히 표현하면, HTML과 같은 마크업 언어를 정의하여 특정 목적에 맞게 사용할 수 있도록 지원한다. 즉, 데이터의 의미와 구조를 표현하기 위한 태그들을 정의할 수 있어서, XML문서를 수신한 브라우저는 데이터의 의미를 이해하고 어떻게 처리해야 할지를 이해할 수 있다. 일례로, 상품 카타로그를 사용자에게 전달하기 위해서는 상품명과 가격을 `<Product Name="TV" Price=10000 />`과 같은 XML표현을 사용할 수 있다. XML은 [그림 1]과 같이, 데이터 구조에 대한 정보를 DTD (Document Type Definition)로 기술할 수 있다. 따라서 이 메타정보를 이용하여 문서내 데이터의 구조적 완전성을 검사할 수 있다. XSL (XML Style Sheet) 을 사용한 XML문서의 구조적 변환도 가능해서, [그림 1]과 같이 XML정보를 웹 브라우저로 검색하기 위해서 HTML문서로 변환하는 것이 가능하다. 또한 다른 형식으로도 변환하여 사용할 수 있다. [표 1]은 XML, SGML, HTML을 상대적으로 비교한 것이다.

### III. XFDL(Extensible Forms Description Language)

문서 양식은 기업간 거래를 위해서 매우 중요한 사용자 인터페이스이다. 분석가들의 추정치로는 매일 약 900억 장의 양식이 작성되며, 이중 상당한 부분이 종이를 사용한 문서 양식이라고 한다. 양식이 이렇게 광범위하게 사용되는 이유를 알기 위해서는 종이 양식이 현재 조직내에서 수행하는 기능을 분석해 볼 필요가 있다.

종이 양식은 특정한 데이터 집합을 분류하고 조직화하여 사용자가 쉽게 이해할 수 있도록 해준다. 데이터를 표현하는 구조화된 방법이 없다면 대용량의

거래 프로세스들은 구현하거나 조직화하기가 매우 어려울 것이다. 구매부서는 일반적으로 중요한 정보를 수집하고 처리하기 위해서 구조화된 양식 데이터에 의존한다. 종이 양식은 구조화된 데이터를 정보 시스템에 제공하기 위한 단순하면서도 훌륭한 방법을 제공한다. 데이터 수집을 위한 템플릿과 수집된 거래 데이터 레코드를 제공한다. 점에서 종이 양식은 독특하다. 대부분의 종이양식은 명시적이거나 암시적으로 계약에 관한 내용을 다룬다. 실제로 사용자들은 필요한 정보를 기입하고 서명함으로써 자신이 계약에 참여함을 명시한다. 만약 문제가 발생하면, 종이 양식은 증거로서 사용된다.

종이 양식은 분명히 거래 프로세스의 핵심적인 부분이지만, 종이 기반의 시스템에 의존하는 것은 효율적이지 못하며 많은 비용이 소요된다. 기업간 전자상거래가 도입되면서, 구매회사와 판매회사는 자신들의 거래 프로세스를 통합하기 위한 방법을 탐색하고 있다. 이러한 유형의 통합에는 XML이 새로운 핵심 기술로 인식되고 있다. 구조화된 데이터를 교환하기 위해서는 데이터 구조를 생성하고 교환하기 위한 표준 문법이 필요하며 XML이 이러한 역할에 적합하다고 할 수 있다. 그러나 XML은 솔루션이 아니라 단지 프레임워크만을 제공한다.

웹기반의 폼과 거래 처리를 위한 많은 접근법이 있으나, 웹상에서의 거래 레코드에 관한 핵심을 이해하기 위해서는 HTML 양식을 살펴보는 것이 좋다. B2C에서는 HTML양식으로 종이 양식을 대신하는 경우가 많다. 비록 HTML 양식이 사용자 인터페이스를 제공하지만, 문서 양식의 두가지 중요한 기능 (데이터 구조의 제공과 데이터 레코드의 생성)을 대신하지는 못한다.

XFDL 4.0은 인터넷상에서 종이 양식을 표현하는 문제를 해결하기 위해서 설계된 고도로 구조화된 XML기반 프로토콜이다. 이 언어의 특징을 정리하면 다음과 같다.

- XFDL은 문서중심의 마크업 언어이다. XFDL은 거래 레코드에 필요한 모든 요소들을 하나의 안전한 파일에 저장하여 디지털 양식을 관리하는 문서중심의 접근법을 채택하고 있다. 이 접근법이 XML 디자인 원칙으로는 논란이 있을 수 있지만, 부인방지를 위한 거래 레코드를 생성하기 위해서 필수적이다. 종이 양식과 같이 XFDL 양식은 내용, 문맥, 구조를 모두 저장하고 있으며, 거래에 관한 논쟁이 발생하면 XFDL 문서가 즉시 검색되어 거래의 실상을 증명하는데 사용되게 된다.
- XFDL은 계산 언어이다. 대부분의 XML언어와는 달리, XFDL은 프로그래밍 언어이기도 하다. XFDL은 의사결정을 하고, 수식을 다루며, 사용자의 입력에 반응할 정도로 지능적이다. XFDL의 세밀한 계산능력은 자동적으로 수학과 논리 함수들의 수행, 사용자 인터페이스 안내, 오류 수정 등의 기능을 허용한다. 이러한 계산 논리는 XFDL문서에 내장되며, 디지털 서명된 거래 레코드의 일부가 된다. 내장된 논리는 외부의 스크립트등과의 연결을 불필요하게 하며, 외부와 단절된 이동 단말 등에서도 사용될 수 있는 장점을 제공한다.
- XFDL은 단언 (Assertion) 기반의 언어이다. XFDL의 계산 논리는 단언을 사용하여 표현된다. 순차적인 방식으로 기능하는 대부분의 프로그래밍 언어와는 달리, XFDL에서는 관리되어야 할 수행 순서가 없다. XFDL에서의 계산의 실행은 표계산 소프트웨어를 사용하는 것과 유사하다. 예를 들면, 필드1이 필드2와 필드3의 합으로 정의되는 경우, 필드1의 값은 필드2와 필드3의 값이 변함에 따라서 자동으로 수정된다. XFDL이 단언기반의 언어로 탄생한 이유로는 2가지를 들 수 있다. 그 중 한가지는 사람들의 독해력을 위해서다. 복잡한 양식에서 사용되는 논리는 단언을 사용하여 표현하면 매우 쉽다. 또 다른 한가지는 사용자가 저장하거나 서명하거나 송신하고자 결정하는 시점에서 문서의 정확한 상태를 정리하기가 매우 쉽다는 점이다. 순차적인 언어에서 정확한 프로그램 상태를 파악하는 것은 많은 양의 정보를 획득해야 하며, 획득된 정보의 해석도 매우 어렵다.
- XFDL은 사람이 독해 가능한 평문이다. XML의 핵심 디자인 원칙중의 한가지는 사람이 읽고 이해할 수 있어야 한다는 점이다. XFDL도 이러한 원칙을 따른다. XFDL은 어떤 인터넷 기술보다도 수명이 긴 거래 레코드를 생성하기 위해서 설계되었다. 이러한 레코드는 몇 년 동안 거래 프로세스의 증거로서 유지할 필요가 있으며, 투명하지 못한 이진 데이터 형식으로 저장하는 것은 바람직하지 않다.
- XFDL은 XML기반의 개방형 표준이다. 사람이 읽고 이해할 수 있도록 설계된 XFDL은 공개적으로 접근 가능한 개방형 표준이 되어야 한다. 왜냐하면, XML엘리먼트의 의미가 널리 알려져 있는 것이 필요하기 때문이다. 이렇게 되면, XFDL문서에 수집된 구조화된 정보는 번역과 정없이 다른 기업과의 공유가 가능해진다. 이러한 기능은 완전히 다른 종류의 시스템에 의존하는 독립적인 기업내부 부서간에도 유용하다.
- XFDL은 확장가능하다. 대부분의 XML기반 언어는 스스로 확장가능한 것이 아니라, 한정된 키워드, 규칙등의 집합으로 이루어진다. 일반적으로 거래의 수명주기는 매우 복잡하며, 양식 브라우저 이외의 프로그램에 의한 처리가 필요하다. XFDL에 특정한 함수를 추가하거나 XFDL 계산 엔진을 양식 브라우저 이외의 프로그램에서 사용하는 능력은 거래 레코드를 관리하는 데 있어서 매우 중요하다. 사용자가 정의한 함수나 프로그램 아이템이 양식과 데이터 베이스 간의 복잡한 상호작용을 필요로 하는 경우나, 양식의 내용에 기반한 업무 흐름의 기술

등이 여기에 해당된다. 실제로 전자상거래에서 대부분의 기본적인 기능들은 서버측에서 수행 되는데, XFDL의 계산 엔진은 거래 문서의 처리능력을 강화하여 이러한 기능들을 지원한다.

[그림 2]는 XFDL 양식 브라우저로 피타고라스 정리를 처리하는 간단한 사례를 보여준다. [그림 2]와 같은 양식은 [그림 3]과 같은 XFDL문서로부터 작성된 것이다. XFDL양식의 최상위 엘리먼트는 XFDL로서 version속성을 갖는다. 이 속성은 반드시 사용되어야 하며, 양식에서 따르는 XFDL 버전을 지칭한다. XFDL 엘리먼트는 복수의 page 엘리먼트를 포함할 수 있다. page 엘리먼트 내에는 버튼 (Button), 필드 (Field), 라벨 (Label), 체크박스 (Checkbox), 팝업 (Pop-up) 등의 다양한 그래픽 항목뿐 아니라 디지털 서명이나 사용자가 정의한 항목 등 보이지 않는 객체들도 포함될 수 있다. 각각의 페이지와 그래픽 항목은 sid (Scope

Identifier) 라고 하는 속성을 반드시 기술하여야 한다. 계산 시스템을 지원하기 위해서 각각의 XML 엘리먼트는 문서내에서 유일한 ID를 보유해야 한다.

첫번째 page 엘리먼트 앞이나, 각 페이지의 첫번째 항목 앞에는 전역적으로 사용되는 조건을 포함할

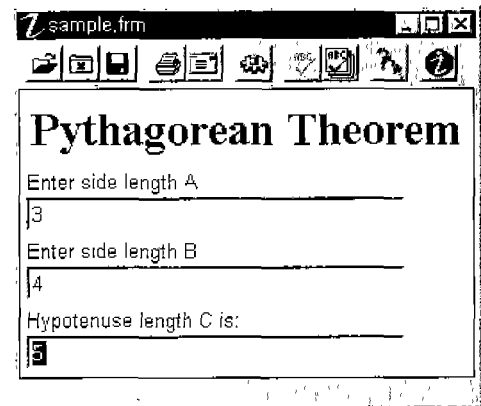


그림 2. XFDL문서 브라우저

```
<?xml version="1.0"?>
<XFDL version="4.0.1">
  <page sid="p1">
    <label sid="Title">
      <value>Pythagorean Theorem</value>
      <fontinfo content="array">
        <facename>Times</facename>
        <ae>18</ae><ae>bold</ae>
      </fontinfo>
    </label>
    <field sid="A"><label>Enter side length A:</label><value>3</value></field>
    <field sid="B"><label>Enter side length B:</label><value>4</value></field>
    <field sid="C">
      <label>Hypotenuse length C is:</label>
      <editstate>readonly</editstate>
      <value content="compute">
        <cval>5</cval>
        <compute><![CDATA[A.value <= "0" || B.value <= "0" ?
          "" : sqrt(A.value*A.value + B.value*B.value)]]>
        </compute>
      </value>
    </field>
  </page>
</XFDL>
```

그림 3. XFDL사례

수 있다. 이러한 조건은 각각의 항목에 대한 기본 설정을 제공한다. 이러한 조건의 구조는 항목에 사용되는 조건의 구조와 동일하다. 라벨과 같이 간단한 항목이라 할지라도, [그림3]의 Title라벨과 같이 값이나 폰트 등과 같은 여러 가지 조건을 포함한다. 조건에 대해서는 sid속성이 요구되지 않는다. fontinfo엘리먼트와 같이 하위 엘리먼트를 추가적으로 포함하는 조건은 배열 (Array) 이라는 XFDL의 개념을 사용한다. ae엘리먼트는 배열 엘리먼트에 의한 사용을 위해서 예약된 엘리먼트이며, 다른 이름의 엘리먼트도 사용된다. 배열 엘리먼트는 계산에서 순차적인 위치로서 참조된다. 예를 들어 보면, p1.Title.fontinfo[facename] 이라는 참조는 Times라는 값으로 평가되는데, p1.Title.fontinfo(0)와 동일하다.

계산되는 엘리먼트는 cval과 compute라는 하위 엘리먼트를 포함하는데, cval의 값은 compute 엘리먼트의 계산식에 의해 계산된 값이다. cval의 값은 디지털 서명이 적용된 뒤에는 고정되게 되며, 거래 부인 방지에서 중요한 역할을 하게 된다. XFDL의 계산식은 3개로 이루어진 의사결정 연산자 (? :) 뿐 아니라 표준적인 연산자들을 지원한다. 6가지의 비교 연산자와 3개의 논리 연산자를 사용할 수 있으며, 중첩된 의사결정 논리도 허용된다. 추가적으로 XFDL은 숫자, 재귀, 문자열 함수를 제공한다. 논리 연산자 &&나 비교 연산자 <는 XML에서 특별한 용도로 사용되기 때문에 계산식에서 사용하기 위해서는 계산식 전체를 CDATA섹션으로 만들

어야 한다.

## IV. XForms

XForms는 HTML 양식을 계승하는 것으로 데이터와 논리를 사용자 인터페이스로부터 분리시켜 동일한 데이터를 여러 종류로 표현될 수 있도록 허용한다. XForms 1.0은 현재 W3C 초안 상태이며, XHTML [6] 과의 통합을 위한 모듈로서 설계되고 있다. 그러나 XForms의 설계는 다른 XML 표준과의 통합을 배제하지 않는다. XForms 양식은 명시적인 데이터 모델, 사용자 인터페이스, 그리고 XML과 유니코드의 사용으로 구성된다. 명시적인 데이터 모델은 데이터간에 제약이 있는 복합 데이터 유형으로 양식을 정의한다. 사용자 인터페이스는 데이터 모델에 종속하는 표현 컨트롤로 정의된다. 서버와의 데이터 교환을 위해서는 XML과 유니코드가 사용된다.

양식에는 사용자에게 대한 질문이나 선택 가능한 조건들, 논리적인 순서 등과 같은 목적 (Purpose) 과 화면상의 표현 (Presentation) 간의 분명한 구분이 필요하다. 양식의 목적은 다양한 미디어나 단말의 특성을 반영하여 기술될 수 있으나, 표현은 최소의 공통 분모에 맞추다 보면 풍부한 정보를 상실하게 된다. XForms에서는 목적과 표현을 각각 데이터 모델과 사용자 인터페이스 규격서로 작성하고 있다. 데이터 모델은 양식의 개념적인 구조를 사용자

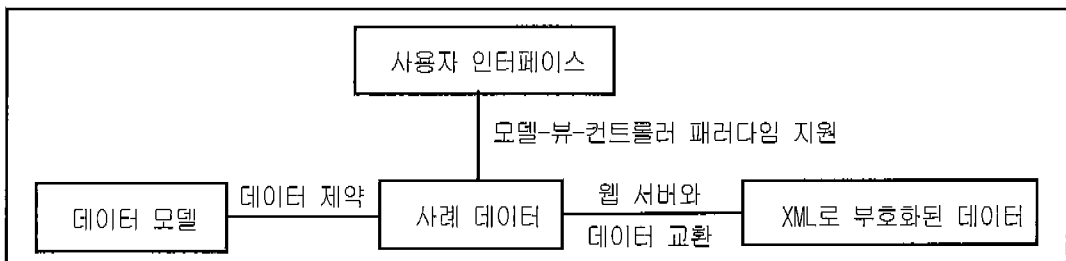


그림 4. XForms 프레임워크

인터페이스와 무관하게 정의할 수 있도록 허용한다. 풍부한 상호작용을 위해서 사용자 인터페이스 계층을 도입할 계획이지만 장치 독립성은 기존 HTML 양식의 기능을 희생시키지 않도록 제한될 것이다. 데이터 모델은 장치 독립적이기 때문에 WML이나 SMIL 등의 다른 XML 표준과의 결합도 가능하다. [그림 4]는 XForms의 프레임워크를 보여준다.

일반적으로 HTML 양식은 데이터베이스나 업무 흐름(Workflow)에 관한 응용의 일부분으로 사용된다. 정보를 데이터베이스로부터 HTML 양식으로 이동시키거나 반대로 양식으로부터 데이터베이스에 제출하는 과정은 양식을 사용하는 여러 시나리오 중 하나에 해당된다. XForms 디자인은 양식 그 자체와 응용과의 일치성을 개선하는데 초점을 맞추고 있다. XML은 메타 수준의 데이터 형식으로 거의 모든 데이터를 표현할 수 있다. 양식이란 하나의 구조화된 데이터 교환이기 때문에 XML은 양식을 완벽하게 소화할 수 있다. [그림 5]는 간단한 형식의 구매 주문서를 XML 문서로 표현한 것이다. XForms의 목적은 [그림 5]와 같은 내용을 양식을 통해서 편집할 수 있도록 지원하는 것이다. 이러한 목적을 달성하기 위해서는 [그림 5]와 같은 XML 문서에 대응하는 XForms 데이터 모델을 구성하는 것이 필요하다. 이러한 과정은 단순하게 수작업으로 수행될 수 있으며, XSLT와 같은 보다 강력한 도구의 지원을 받을 수도 있다. [그림 5]로부터 구성된 데이

```
<?xml version="1.0"?>
<purchaseOrder>
  <shipTo>
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
</purchaseOrder>
```

그림 5. 구매주문서의 XML 표현

```
<group name="purchaseOrder">
  <group name="shipTo">
    <string name="name"/>
    <string name="street"/>
    <string name="city"/>
    <string name="state"/>
    <string name="zip">
      <mask>dddd</mask>
    </string>
  </group>
</group>
```

그림 6. 구매주문서 작성을 위한 XForms 데이터 모델

터 모델은 [그림 6]과 같은 형태로 구성될 수 있다.

양식을 표현하는 XForms 데이터 모델로부터 정규형의 XML 문서가 생성된다. 이 때, 데이터 모델의 종단(Leaf) 노드 각각에 대해서는 노드의 name 속성 값과 동일한 엘리먼트를 생성하고 데이터의 값이 이 엘리먼트의 내용이 된다. 마찬가지로, 중간 노드 각각에 대해서는 데이터 그룹의 name 속성 값과 동일한 엘리먼트를 생성하고, 데이터 그룹에 포함된 하위 엘리먼트들이 이 엘리먼트의 내용이 된다.

XForms 데이터 모델과 사례 데이터는 사례 문서에 포함될 수 있다. 사례 문서에서 XForms 데이터 모델과 사례 데이터는 [그림 7]과 같이 xform 엘리먼트에 model 엘리먼트와 instance 엘리먼트로 각각 첨부될 수 있다.

[그림 7]에서는 XHTML 표준의 양식 엘리먼트들을 사용자 인터페이스 용도로 사용하고 있다. XForms 데이터 모델과 사례 데이터를 XHTML 양식 엘리먼트와 연결시키기 위해서 XHTML 상의 form 이름은 xform 엘리먼트의 id 속성 값으로 설정된다. 그리고 XHTML 양식 컨트롤의 name 속성은 데이터 모델의 해당되는 필드 절대 경로를 설정해 준다. 예를 들면, street 컨트롤을 위한 필드는 purchase-Order.shipTo.street로 설정된다.

사용자가 입력 필드의 값을 변화시키면, 사례 데



```

<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML-XForms 1.0//EN"
"http://www.w3.org/TR/xhtml-forms1/DTD/xhtml-xforms1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Purchase Order</title>
    <xform xmlns="http://www.w3.org/2000/xforms"
      action="http://www.my.com/cgi-bin/receiver.pl"
      method="postXML"
      id="po_xform">
      <model>
        <group name="purchaseOrder">
          <group name="shipTo">
            <string name="name"/>
            <string name="street"/>
            <string name="city"/>
            <string name="state"/>
            <string name="zip">
              <mask>ddddd</mask>
            </string>
          </group>
        </group>
      </model>
      <instance>
        <purchaseOrder>
          <shipTo>
            <name>Alice Smith</name>
            <street>123 Maple Street</street>
            <city>Mill Valley</city>
            <state>CA</state>
            <zip>90952</zip>
          </shipTo>
        </purchaseOrder>
      </instance>
    </xform>
  </head>
  <body>
    <h1>Shipping Information</h1>
    <form name="po_xform">
      Name: <input name="purchaseOrder.shipTo.name"/><br/>
      Street: <input name="purchaseOrder.shipTo.street"/><br/>
      City: <input name="purchaseOrder.shipTo.city"/><br/>
      State: <input name="purchaseOrder.shipTo.state"/><br/>
      Zip: <input name="purchaseOrder.shipTo.zip"/><br/>
      <button onclick="submit('po_xform')">Submit</button>
    </form>
  </body>
</html>

```

그림 7. XHTML문서와 데이터모델의 결합

이터는 갱신되며, submit버튼을 클릭하면, 사례 데이터는 서버로 전송된다. 서버측에서는 이렇게 전송된 XML문서를 스키마를 사용하여 검증하고, 기존의 문서를 새로운 문서로 대체한다.

### V. 결 론

기업간 전자상거래를 위한 양식관리는 XML기반의 XForms를 중심으로 전개될 것으로 보인다. 현재 사용자 인터페이스 부분에 대한 XForms 작업이 미진하기 때문에, 웹상에서의 완벽한 양식관리를 위해서는 시간이 필요하다. 사용자 인터페이스 부분이 완성되기까지는 기존의 웹 컨트롤을 사용하여 데이터 모델의 제약을 구현하는 방안도 적용될 수 있다.

[그림 8]은 XSheet 양식 디자이너 [9] 로서 XML 스키마로 표현된 문서의 내용을 바탕으로 문

서의 양식을 설계하기 위한 그래픽 도구이며, 하나의 내용 문서에 대하여 복수의 양식을 설계할 수 있다. 설계된 내용은 XSLT로 출력되며, XML로 표현된 데이터에 적용되어 HTML기반의 양식을 생성하게 된다. 중요한 점은, 양식의 입력이 완성되면, 입력으로부터 문서를 구성하는 스크립트 프로그램이 작동되며, 서버에는 이렇게 생성된 XML문서가 전달된다. 이러한 XML기반의 그래픽 도구들은 전자상거래 양식의 생성 및 관리를 획기적으로 개선할 수 있으며, XForms의 수용도 용이하게 가능하다.

현재 XForms 데이터 모델과 XML 스키마 [7]의 데이터 유형이 서로 중첩되는 부분이 많기 때문에 조정이 필요하다. 이에 대한 대안으로는 XSLT [5]를 적용하여 XML 스키마로부터 XForms 데이터 모델을 생성하는 방법이 있다. 이 방법은 데이터 모델과 사용자 인터페이스의 독립적인 사용을 지원하기 때문에 다양한 사용자 인터페이스로의 변환을 지원하게 될 것이다.

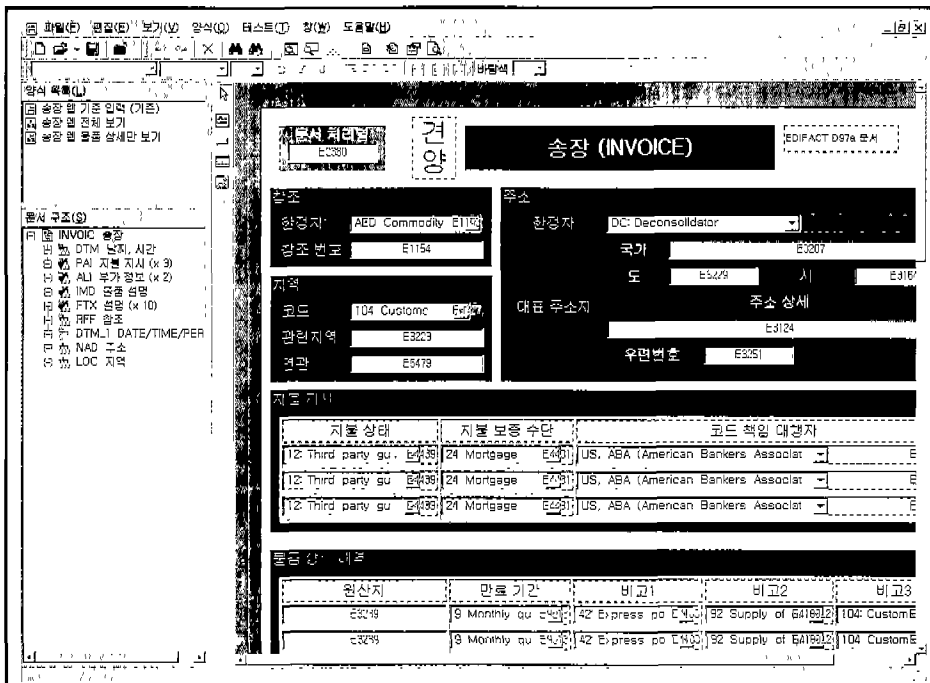


그림 8. XSheet 양식 디자이너

## \*참고문헌

- [1] Blair, B.T. and J. Boyer, "XFDL: Creating Electronic Commerce Transaction Records Using XML," <http://www8.org/w8-papers/4d-electronic/xfdl/xfdl.html>, The Eighth International WWW Conference, May 1999.
- [2] Cover, R., "The XML Cover Pages: XFDL," <http://www.oasis-open.org/cover/xfdl.html>, April 2000.
- [3] W3C, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," <http://www.w3c.org/TR/REC-smil/>, June 1998.
- [4] W3C, "Extensible Forms Description Language (XFDL) 4.0," J. Boyer et al. (Editors), <http://www.w3c.org/TR/1998/NOTE-XFDL-19980902>, September 1998.
- [5] W3C, "XSL Transformations (XSLT) Version 1.0," <http://www.w3c.org/TR/xslt/>, November 1999.
- [6] W3C, "XHTML 1.0: The Extensible HyperText Markup Language," <http://www.w3c.org/TR/xhtml1>, January 2000.
- [7] W3C, "XML Schema Part 0: Primer," <http://www.w3c.org/TR/xmlschema-0/>, April 2000.
- [8] W3C, "XForms 1.0: Data Model," M. Dubinko et al. (Editors), <http://www.w3c.org/TR/xforms-datamodel/>, April 2000.
- [9] 김형도, "An XML-Based Form Management System for B2B Electronic

Commerce," Working Paper, 2000.



김 형 도

1985년 서울대학교 산업공학과 졸업 (학사)  
 1987년 한국과학기술원 경영학과 졸업 (석사)  
 1992년 한국과학기술원 경영학과 졸업 (박사)  
 1992년 한국과학기술원 경영학과 대우교수  
 1993년~1999년 (주)데이콤 EC인터넷 기술팀장  
 1999년~2000년 울산대학교 객원교수, (주)DIB기술  
 고문  
 2000년~현재 아주대학교 정보통신 전문대학원 교수,  
 ebXML 전문위원  
 관심분야: 전자상거래, XML, Data Mining, 시스템 모  
 델링/시뮬레이션, 3D 게임