

XML 스키마 발전 동향

황병연*, 김연혜**

● 목 차 ●

1. 서론
2. XML DTD의 단점
3. DTD를 보완한 XML 스키마
4. XSD
5. 결론

1. 서론

XML(eXtensible Markup Language) 문서는 적격 문서(Well-Formed Document)와 유효한 문서(Valid Document)로 나눌 수 있다. 적격 문서란 XML 문서에 사용된 모든 요소들이 시작 태그와 끝 태그를 가지고 있고, 중첩 규칙을 위반하지 않는 문서를 의미하는 것으로 스키마(schema)가 없는 XML 문서를 비 검증용 파서(nonvalidating parser)로 파싱했을 때 오류가 없는 문서를 말한다. 따라서, 적격 문서는 XML 사양에 명시되어 있는 XML 문서로서 간주되는데 필요한 최소한의 필수 조건을 만족해야 한다. 유효한 문서란 XML 문서가 스키마의 정의대로 올바르게 작성되었는지 검사하는 유효성 검증(validating) 과정을 거치게 되는데, 스키마가 있는 XML 문서를 검증용 파서(validating parser)를 사용하여 파싱했을 때 오류가 없는 문서를 말한다. 따라서, 유효한 문서는 적격 문서보다 더 엄격한 조건을 정확히 지켜야만 한다[1].

XML 문서에서 SGML(Standard Generalized

Markup Language)처럼 DTD(Document Type Definition)를 기본으로 하여 유효한 문서만을 허락하지 않고 적격 문서도 허락한 이유는 모든 XML 문서를 한번 읽을 때마다 문서의 유효성을 검증하는 데 시간을 소비하는 것을 방지하고 개인적으로 마크업을 자유롭게 구성하여 XML 문서를 만드는 것을 지원하기 위해서이다. 그러나 다른 사람과 정보를 공유하기 위해서는 문서의 구성 방법과 마크업 방법을 정확히 규정해야 한다. 왜냐하면 XML 문서의 애플리케이션 같은 정확한 구조를 예상해야 하는 분야에서는 안정성과 신뢰성이 고려되어야 하기 때문이다. 따라서 적격 문서보다 더 엄격한 조건을 적은 문서를 스키마 문서라고 한다.

스키마의 사전적인 의미는 일반화된 계획이나 도해를 뜻하지만 XML의 경우 스키마란 데이터가 마크업 되는 방식으로서 문서 안에서 어떤 요소형과 특성, 값들을 사용할 수 있는지에 대한 규정이라고 할 수 있다. 즉, 하나의 XML 문서 안에 어떤 것들을 담을 수 있는지에 대한 규칙들의 집합이며 XML Vocabularies 또는 XML Dictionaries라고도 불린다. 그러나 DTD의 단점을 보완한 제안들(예를 들어 XML-Data Schema, Schema for Object-oriented XML)과 XSD(XML Schema Definition) 등 용어로서

* 가톨릭대학교 컴퓨터전자공학부 교수

** 가톨릭대학교 컴퓨터공학과 석사과정

의 스키마가 사용되어짐으로써 DTD와 스키마는 별개의 개념으로 사용되어졌다. 즉, 사전적인 의미로서의 스키마에는 DTD가 포함되고, 용어로서의 스키마는 DTD를 보완한 것으로 다양한 데이터 형식과 이름 공간(namespace) 등을 허용하고 유연한 문서 구조를 가지도록 지원하여 데이터 중심의 XML 문서와 애플리케이션에 유효성을 보장시키는 DTD의 대안이다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 사용중인 DTD의 단점을 기술하고, 3장에서는 DTD의 단점을 보완하여 W3C(World Wide Web Consortium)에 제출된 스키마관련 제안들을 제시한다. 4장에서는 XML 2.0을 위해서 2001년 5월 1일에 W3C에서 표준으로 승인된 XSD에 대해 소개한 후 5장에서 결론을 맺는다.

2. XML DTD의 단점

XML 1.0에서는 XML 문서의 타입을 정의하기 위한 메커니즘(DTD)을 제공하지만 XML 개발자들은 DTD를 구현한 경험에 의해 DTD의 수많은 단점과 내용 모델을 정의하는 더욱 포괄적이고 엄격한 방법이 필요하다는 것을 알게 되었다. DTD의 대표적인 단점은 다음과 같다[2].

- ① DTD를 기술하는 문법은 XML을 기술하는 문법과 다르다. DTD의 경우 EBNF(Extended Backus-Naur Form)표기법을 사용함으로써 XML 문서와 달리 사람이 읽는 데 어려움이 많다. 또한 개발자가 XML 문법과 DTD 문법 둘 다 알아야 하는 어려움과 DTD를 작성하는 소프트웨어와 XML문서를 작성하는 소프트웨어를 분리시켜야 할 뿐 아니라 DTD 문서와 XML 문서를 구문 분석하는 파서 역시 분리시켜야 한다.
- ② 확장성을 갖추지 못한다. DTD를 갱신했을 때, 이전의 DTD로 검증된 모든 문서들을 갱

신된 DTD로 다시 검증해야만 한다. DTD를 변경하지 않고서는 XML 문서를 확장시킬 수 없다.

- ③ XML 문서의 데이터 타입이 문자 타입인, #PCDATA(Parsed Character DATA)와 CDATA(Character DATA) 둘 뿐이므로 제대로 표현하지 못한다.
- ④ DTD는 상속과 하위 분류에 제한이 있어 객체 지향 디자인에 부응하지 못한다.
- ⑤ DTD는 이름 공간을 지원하지 못한다. 컴퓨팅 환경이 XML 중심으로 변모하고 XML 정보간의 상호 연동이 확산되면 될수록 이름 공간의 필요성이 높아지고 있는 반면에 DTD는 효용가치가 떨어지기 때문이다.
- ⑥ DTD만으로는 기술 능력의 한계 때문에 강력한 설명을 지원할 수 없다.
- ⑦ 기본적인 요소내용에 대한 기능이 없다.
- ⑧ DTD는 주어진 요소나 속성이 어떤 종류의 정보를 포함하고 있는가를 제어할 수 없다.
- ⑨ 제대로 된 DTD 작성이 어렵다.
- ⑩ DTD에 있는 모든 선언들은 포괄성을 가져 다른 문맥에 있다하더라도 같은 이름을 사용할 수 없다.

이러한 단점들을 보완하기 위한 해결책으로 다음 장에 나올 W3C에 제출된 스키마 관련 노트(Note)들에 제안되었고 그 결과 XSD가 새로운 XML 스키마 표준안으로 제정 중에 있다.

3. DTD를 보완한 XML 스키마

앞장에서 기술한 DTD의 단점을 보완하는 방안으로 많은 스키마 관련 제안들이 제시되었다. DT4DTD(Datatypes for DTDs)를 제외한 나머지 제안들은 DTD를 보완하는 스키마를 제안한 것이고 DT4DTD의 경우 문자 타입의 데이터 타입을 지원

하는 DTD의 다양한 데이터 타입을 지원하는 제안이다. DTD를 보완하는 제안들과 표준화 중인 XSD의 장점은 XML 문법과 동일하다는 것과 다양한 데이터 타입을 사용할 수 있다는 것이다. XML 문법을 채택한 것의 장점은 XML 문서의 파서와 애플리케이션을 스키마 문서에 그대로 적용할 수 있다는 것과 확장성과 유연성, 학습의 용이성을 가진다는 것이다.

이러한 스키마들이 유효성을 보장하는 방식은 스키마 문서 자체가 XML 문서가 되어 XML 문서처럼 스키마 문서도 DTD가 존재함으로써 유효성이 보장되고 그 스키마 문서를 사용한 XML 문서의 유효성도 보장되는 방식이다. 다음에 나온 <표 1>의 제안들은 XML 애플리케이션들 간의 정보 교환을 위한 언어로 자리 잡게 하는 데 매우 중요한 역할을 하였다.

<표 1> XML 스키마 제안이 제출된 시기

Notes	W3C에 제출된 시기
XML-Data	1998/01/05
DCD	1998/07/31
SOX	1998/09/30 (for XML 1.0)
	1999/07/30 (for XML 2.0)
DDML	1999/01/19
DT4DTD	2000/01/13

3.1 XML-Data Schema

Microsoft, ArborText, DataChannel, Inso에서 제안한 XML-Data Schema[3]는 다음과 같은 특징을 갖는다.

- ① 단지 문서의 구문 구조를 기술하는 구문 스키마를 가지는 DTD와 달리 XML-Data는 개념이나 객체들의 관계를 기술하는 개념 스키마를 보강한 제안이다.
- ② XML-Data에서는 다양한 데이터형을 지원한다.
- ③ 이름 공간을 지원한다.

- ④ 별명, 요소와 속성의 상관관계 및 계통구조, 참조, 다중 키 개념을 이용하여 개념스키마를 구현한다. 또한 그룹 모델, 제한규칙, 요소 내용의 기본값 설정도 가능하도록 지원한다.
- ⑤ 열린 모델을 지원하여 XML문서가 유효한 문서이면서 확장될 수 있도록 지원한다.

이 제안은 W3C에서 제출된 최초의 스키마 제안이며 DTD에서 유효하지 않았던 여러 새로운 데이터형을 제공한다. 따라서 다음에 나올 많은 제안들은 이 제안의 부분집합을 포함한다.

이 제안을 한 컨소시엄 중에 하나인 Microsoft에서 간소화한 XML DR(XML Data Reduce)을 BizTalk[4]에서 실제로 사용하고 있으며 브라우저인 인터넷 익스플로러 버전 5.0에서부터 지원을 하고 있다. 또한 스키마 제안으로서 끝난 것이 아니라 웹 기반의 메타 데이터 기술과 교환을 위한 구조로 W3C에서 표준화하고 있는 RDF(Resource Description Framework)[5]의 일부가 되었다.

3.2 DCD(Document Content Description for XML)

IBM, Microsoft에서 제안한 DCD[6]는 RDF의 구문 구조를 사용한 구조적 형식에 대한 모델로써 XML문서의 내용과 구조에 대한 제한 사항들을 기술한 하나의 RDF vocabulary이며 RDF를 만족하는 방법으로 XML-Data 부분집합을 포함시킨 절충안이다.

또한 XML-Data에서 제안한 콘텐츠 모델인 개방형과 폐쇄형을 DCD에서도 사용하여 기존 규칙에 대해 유효한 문서를 유지하면서 미래의 확장안을 추가할 수 있게 하였다.

DCD를 설계한 원칙은 다음과 같다.

- ① DTD의 어의(semantics)를 포괄하는 의미론을 가져야 한다.
- ② DCD 데이터 모델 및 문법은 RDF의 모델과

문법에 적합해야 한다.

- ③ 하나의 DCD에 담긴 제한 조건들은 저작도구나 문서의 내용과 구조에 관한 정보를 얻고자 하는 기타 응용프로그램들이 직접적으로 사용할 수 있어야 한다.
- ④ 다른 W3C 워킹 그룹들이 제시하는 메커니즘들이 적절하고 효율적이라면 DCD는 그것들을 수용해야 한다.
- ⑤ DCD는 사람이 읽을 수 있으며 논리적으로 명확해야 한다.

위의 두 번째 설계원칙에 따르면 RDF 구문 구조를 단순화해야 한다. 그래서 RDF 워킹 그룹에 이 에 관련한 제안을 하였고, 이것을 RDF 워킹 그룹에서 비준했을 때에만 RDF 구문 구조를 DCD에서 사용할 수 있다는 제약을 가지게 되었다. DCD의 장점으로는 RDF 구문 구조를 이용하여 반복적이지 않은 속성들을 부 요소, 특성으로 표현할 수 있다는 것과 description이라는 부분요소를 사용하여 사람이 요소의 목적을 쉽게 이해하도록 돕는다는 것이다.

DCD와 전에 나온 스키마와 동일한 점은 하나의 문서 안에서 어떤 요소형과 특성, 값들을 사용할 수 있는 지에 대한 규정이라고 할 수 있고, 다른 점은 제안 문서 내에서는 문서의 구조와 내용을 제한, 강제하는 것이라고 강조하고 있다. DCD가 지원하는 데이터형은 SQL과 최신 프로그래밍 언어들이 지원하는 데이터형에 기반하고 있으며, DCD에 참여한 Microsoft는 브라우저인 인터넷 익스플로러 5.0에서 XML-Data와 XML-DR과 함께 DCD도 적용할 수 있게 만들었다.

3.3 SOX(Schema for Object-oriented XML)

Veo Systems은 1998년에 XML 1.0을 위한 제안을 했고, 1999년에는 XML 2.0을 위한 제안을 제출했

다. SOX[7]도 DTD에 의해서 유효성을 보장받으면서 DTD의 단점을 모두 보완한다.

SOX의 가장 중요한 설계 관점은 XML 문서의 소프트웨어를 더 효율적으로 개발하도록 지원하는 것이다. 또한 이종 애플리케이션들간에 상호운용(interoperability)과 같은 복잡성을 감소시키도록 지원한다. 따라서 XML 문서에 다양한 형태의 내용(멀티미디어 자료) 등을 지원하는 것뿐 아니라 삽입한 XML 문서의 문서화(documentation)와 미리 정의된 다른 스키마들을 관리할 수 있으며 이러한 것들은 강력한 타입 체크와 유효성 검사를 통해서 지원되도록 설계하였다.

처음 제안 환경은 전자상거래 등과 같은 XML 문서의 애플리케이션 스키마 작성시 이전에 만들어진 DTD와 전에 제출된 스키마 제안들(XML-Data, DCD)로 구현된 스키마들을 포함하고, 이전의 제안을 SOX로 변환을 용이하도록 하며 이전에 만든 스키마들을 관리하는 것에 있다. 따라서 SOX는 XML 1.0, XML-Data, DCD로부터 많은 영향을 받았으며 특히 객체 지향 언어인 자바 프로그래밍 언어에서 가장 많은 영향을 받았다.

3.4 DDML(Document Definition Markup Language)

DDML[8]은 이전에 나온 제안들과 달리 XML-Dev 메일링 리스트의 협력자들에 의해 제안되었다. XSchema와 기술(description)적인 차이밖에 없는 DDML은 기존 소프트웨어에 즉시 유용할 수 있게 설계되었고, DDML 문서에서부터 DTD 문서까지의 변환을 정의하고 있다.

DDML은 DTD의 논리적인 내용을 기호화한 것이며, XML 문서 문법, 요소와 속성 중첩과 DDML을 사용하고 있는 프로세서에 의해 검사될지도 모르는 모든 제약 조건을 기술한다.

3.5 DT4DTD

이 전에 소개한 XML-Data를 위시한 스키마 제안들과 XSD는 XML 문서이므로 각각 DTD가 존재한다. 그러나 이 전에 소개한 제안들과 XSD의 다양한 데이터 타입들은 DTD에서는 단지 두 개의 스트링 데이터 타입으로 표현된다. 따라서 스키마 제안들과 다음에 나올 XSD의 경우 DTD로 변환할 때 모든 데이터 타입이 2개로 변환된다. 이를 보완하여 DTD 제작자가 데이터 내용과 속성 값의 다양한

자료 타입을 명세 할 수 있도록 제안한 것이 DT4DTD[9]이다.

4. XSD

DTD의 단점들을 보완하는 방안으로 W3C에 많은 제안들이 올라왔고 마침내 XSD가 XML 2.0을 위해서 2001년 5월 1일에 W3C에서 표준으로 승인 되었으며 XSD는 다음과 같은 특징들을 갖는다.

<표 2> XSD의 간단한 데이터 타입들

Simple Types	length	min Length	max Length	pattern	enumeration	white Space	max Inclusive	max Exclusive	min Inclusive	min Exclusive	total Digits	fraction Digits
string	y	y	y	y	y	y						
normalizedString	y	y	y	y	y	y						
token	y	y	y	y	y	y						
byte				y	y	y	y	y	y	y	y	y
unsignedByte				y	y	y	y	y	y	y	y	y
base64Binary	y	y	y	y	y	y						
hexBinary	y	y	y	y	y	y						
integer				y	y	y	y	y	y	y	y	y
positiveInteger				y	y	y	y	y	y	y	y	y
negativeInteger				y	y	y	y	y	y	y	y	y
nonNegativeInteger				y	y	y	y	y	y	y	y	y
nonPositiveInteger				y	y	y	y	y	y	y	y	y
int				y	y	y	y	y	y	y	y	y
unsignedInt				y	y	y	y	y	y	y	y	y
long				y	y	y	y	y	y	y	y	y
unsignedLong				y	y	y	y	y	y	y	y	y
short				y	y	y	y	y	y	y	y	y
unsignedShort				y	y	y	y	y	y	y	y	y
decimal				y	y	y	y	y	y	y	y	y
float				y	y	y	y	y	y	y		
double				y	y	y	y	y	y	y		
boolean				y	y	y						
time				y	y	y	y	y	y	y		
dateTime				y	y	y	y	y	y	y		
duration				y	y	y	y	y	y	y		
date				y	y	y	y	y	y	y		
gMonth				y	y	y	y	y	y	y		
gYear				y	y	y	y	y	y	y		
gYearMonth				y	y	y	y	y	y	y		
gDay				y	y	y	y	y	y	y		
gMonthDay				y	y	y	y	y	y	y		
Name	y	y	y	y	y	y						
QName	y	y	y	y	y	y						
NCName	y	y	y	y	y	y						
anyURI	y	y	y	y	y	y						
language	y	y	y	y	y	y						
ID	y	y	y	y	y	y						
IDREF	y	y	y	y	y	y						
IDREFS	y	y	y		y	y						
ENTITY	y	y	y	y	y	y						
ENTITIES	y	y	y		y	y						
NOTATION	y	y	y	y	y	y						
NMTOKEN	y	y	y	y	y	y						
NMTOKENS	y	y	y		y	y						

- ① XML-Data와 DCD 제안에서 요소와 요소형 선언, 속성과 속성형 선언을 분리해 놓았던 것과 달리 XSD에서는 통합하여 더욱 간단히 하였다.
- ② 더욱 복잡한 타입 선언이 가능하며 새로운 데이터형을 생성하여 사용할 수 있다.
- ③ 단순한 타입을 사용할 때 페이스(Facets)를 사용하여 제한을 강화하는 동시에 고유한 타입을 만들 수 있다. <표 2>는 XSD에서 사용할 수 있는 단순한 데이터 타입들과 그 데이터 타입들이 사용할 수 있는 페이스들을 소개한다[13].
- ④ 스키마 문서안에 schemaLocation 지시자를 사용하여 또 다른 스키마 문서를 포함시킬 수 있다.
- ⑤ 포괄적 요소들(XML 문서 전체에서 동일하게 사용되는 것)과 지역적 요소들(특별한 문맥에서 고유한 의미를 가질 수 있는 것)을 모두 표현 가능하다.
- ⑥ Annotation이라는 구성요소를 사용하여 사람이나 컴퓨터가 사용할 수 있도록 애플리케이션 정보와 사람의 정보를 가장 중요한 스키마 요소의 시작과 스키마의 최상위 레벨 어딘가에 허용될 수 있도록 규정되어 있다.

5. 결론

지금까지 DTD의 단점을 보완한 여러 스키마 제안들과 2001년 5월 1일에 W3C에서 표준으로 승인된 XSD에 대해서 알아보았다. 앞에서 언급했듯이 XML 문서의 스키마는 SGML의 DTD처럼 강제 사항이 아니다. 문서를 분석할 때마다 문서의 유효성을 확인하기 위한 시간을 소비하여 속도를 저하시키는 스키마는 XML 문서를 다루는 애플리케이션에 따라서 적용할 수도 있고 안 할 수도 있다. 단, 스키마를 적용시키지 않는다면 속도 면에선 이득

을 얻겠지만 다른 XML 애플리케이션과의 통신 등으로 XML 문서들간의 자료들이 상호 교환될 때에는 유효성이 보장되지 않으므로 신뢰성과 이식성에 문제가 생긴다.

스키마를 적용시켜 XML 애플리케이션을 개발하고자 할 때 최근에 표준으로 승인된 XSD는 현재로서는 파서 등의 개발 도구가 부족하여 적용하기가 어렵다. 따라서 사용 가능한 스키마로써 DTD와 Microsoft에서 제안하고 사용중인 XML-DR을 들 수 있다. 이 때, XML-DR이 DTD를 보완한 제안이라고 해서 DTD가 완전히 경쟁력을 잃은 것이 아니며 다음과 같은 장점을 지닌다[14].

- ① XML-DR은 Microsoft에서 제안하여 사용하는 것이므로 MSXML 라이브러리를 사용하지 않으면 사용할 수 없다. 반면에 DTD는 임의의 플랫폼에서도 독립적이므로 이식성이 좋다.
- ② 유효성이 과거나 지금이나 미래에도 보장될 것이다. DTD는 XML 1.0 표준부터 함께 표준화된 것이므로 유효성이 늘 보장된다.
- ③ 표준화된 지 오래되어 많은 에디터들과 검증 파서들이 나와있어 개발환경이 갖추어져 있다.

위와 같은 장점을 가지기는 하지만 DTD의 경쟁력이 떨어지는 것은 확실하다. 그러나 DTD가 완전히 사라질지는 의심스럽다. 이에 반해, 최근 W3C에서 표준으로 승인된 XSD의 경우 파서를 생산하는 업체에서 이것을 채택하여 파서를 생산하고 XSD 관련 애플리케이션 등이 개발된다면 XSD는 XML 문서 형태의 콘텐츠를 정의하기 위한 최상의 선택이 될 것으로 전망된다.

참고문헌

- [1] 채진석, "XML 문서와 DB의 효율적인 연동을 위한 XML 스키마 기술 연구", 한국정보과학회

데이터베이스 연구, 16권 2호, pp.23-28, 2000.

- [2] Frank Boumphrey 외 11인, Professional XML Applications, wrox, 1998.
- [3] <http://www.w3.org/TR/1998/NOTE-XML-data>
- [4] <http://www.microsoft.com/biztalk>
- [5] <http://www.w3.org/RDF>
- [6] <http://www.w3.org/TR/NOTE-dcd>
- [7] <http://www.w3.org/TR/NOTE-SOX>
- [8] <http://www.w3.org/TR/NOTE-ddml>
- [9] <http://www.w3.org/TR/dt4dtd>
- [10] <http://www.w3.org/TR/Schema.html>
- [11] Richard Blair 외 12인, Professional ASP XML, wrox, 2000.
- [12] Elizabeth castro, XML for the World wide web : Visual QuickStart Guide, Peachpit Press, 2001.
- [13] <http://www.w3.org/TR/2001/PR-xmlschema-2-20010330/>
- [14] James Britt & Teun Duynstee, Professional Visual Basic 6 XML, wrox, 2000.

저자약력



황 병 연

1986년 서울대학교 컴퓨터공학과(공학사)
 1989년 한국과학기술원 전산학과(공학석사)
 1994년 한국과학기술원 전산학과(공학박사)
 1999년 2월-2000년 2월 Univ. of Minnesota Visiting Scholar
 1994년-현재 가톨릭대학교 컴퓨터전자공학부 부교수
 관심분야: 공간 데이터베이스(GIS), XML, WWW 데이
 터베이스, 전자상거래 등
 e-mail : byhwang@www.cuk.ac.kr



김 연 혜

2001년 가톨릭대학교 컴퓨터통신학과(공학사)
 2001년-현재 가톨릭대학교 컴퓨터공학과 석사과정
 2001년-현재 (주) 다음기술 연구원
 관심분야: XML, WWW 데이터베이스, 전자상거래 등
 e-mail: yhkim@ntechs.com