Fast Infoset을 이용한 Binary XML Encoder의 설계 및 구현

유성재* · 최일선* · 윤화묵* · 안병호** · 정회경*

*배재대학교 컴퓨터공학과 · **충청대학교 보건의료정보과

Design and Implementation of Binary XML Encoder using Fast Infoset

Seong-jae Yu* · Il-sun Choi* · Haw-mook Yoon* · Byeong-ho Ahn** · Hoe-kyung Jung*

*Dept. of Computer Engineering, Paichai University

**Dept. of Health and Medical Information, Chung Cheong University

E-mail : *{settaire, jkchoi, hkjung}@pcu.ac.kr, *hmyoon@kisti.re.kr, **bhahn@ok.ac.kr

요 약

XML(Extensible Markup Language)은 운영환경에 독립적인 문서형식을 정의할 수 있다는 장점으로 인해 많은 사용자층을 확보하여 현재 가장 널리 쓰이는 문서형식 중의 하나가 되었다. 그러나 이러한 XML이 모바일 분야와 같이 제한된 메모리와 빠른 전송속도를 요구하는 환경에서도 쓰이게 됨에 따라 파일의 용량과 전송속도에 대한 문제점이 새롭게 대두되고 있다. 이에 XML의 바이너리화에 대한 논의가 진행되고 있으며, XML의 구조를 유지하여 기존의 장점을 최대한 살리면서 문서 크기를 줄일 수 있는 방안으로 XML 정보셋을 이용한 Fast Infoset 방식이 주목을 받고 있다.

이에 본 논문에서는 XML을 바이너리화하기 위하여 Fast Infoset 방식 및 ASN.1(Abstract Syntax Notation One)의 인코딩 방법 중 하나인 PER(Packed Encoding Rules)을 이용하여 모듈을 설계하였으며, XML 문서가 바이너리 XML 문서로 인코딩되는 과정을 단계적으로 확인할 수 있는 인터페이스로 인코더를 구현하였다.

ABSTRACT

XML is the most widely used document format by advantage that self-contained for platform. So, currently the most used among other document format. but XML appeared new problem that memory and transmission. And that be used in environment a request restriction memory and fast transmission as like mobile field. Although discussion of XML binarization is going on progress. And Fast Infoset configuration using XML Information Set is receiving attention that a way to lower file size of hold down a existing usage.

In this paper, we designed of module using Fast Infoset and PER among ASN.1 Encoding Rule for XML binarization. And we implementation of encoder constructed interface by stage of translation from XML into binary XML.

키워드

XML, XML Information Set, Fast Infoset, ASN.1

Ⅰ. 서 론

XML 문서는 사람이 읽기 쉬운 형식을 취하고 있고 운영환경에 독립적인 문서형식을 정의할 수 있다는 장점으로 인해 다양한 분야에서 표준으로 자리 잡고 있다. 하지만 이러한 XML은 제한적인 메모리를 가지는 모바일 환경이나 빠른 전송속도를 요구하는 그리드 컴퓨팅과 같은 분야에서 사용되기에는 몇 가지 문제점이 존재한다. XML은 텍스트를 기반으로 이루어져 있으며, 시작과 끝 태그가 모두 붙는 형식을 가지기 때문에 불필요하게 파일크기가 크다는 단점을 가진다. 지금까지는 이를 위해 Gzip과 같은 압축방식이 이용되었으나 이러한 단순 압축 방식들은 XML 문서구조를 고려하지 않은 방식이기 때문에 크기는 줄어질지만 문서를 처리하기에 앞서서 압축과 해제과정이 필요하다는 단점이 있다[1].

이에 XML의 문서 크기를 줄이면서 문서구조를 유지하여 처리 효율에 영향을 미치지 않는 인코딩의 필요성이 높아지고 있다. W3C에서는 바이너리 XML의 필요성 확인을 위한 분석을 진행하는 워킹 그룹(XBC-WG : XML Binary Characterization Working Group)을 통해 바이너리 XML의 효율성을 검증하였으며, 썬 마이크로시스템즈에서도 XML의 바이너리 인코딩에 관한 프로젝트를 진행 중이다. 이 중에서 Sun Microsystems가 진행 중인 프로젝트인 XML 정보셋을 이용한 Fast Infoset 방식이 가장 많은 주목을 받고 있다[2,3].

이에 본 논문에서는 XML의 효율적인 바이너리화를 위하여 Fast Infoset 방식 및 ASN.1의 인코딩 방법 중 하나인 PER을 이용하여 시스템을 설계하였다. 그리고 XML 문서가 바이너리 XML 문서로 인코딩되는 과정을 단계적으로 확인할 수 있는 인터페이스로 인코더를 구현하였다.

II. 관련연구

2.1 XML 정보셋 (Information Set)

XML 정보셋은 XML 문서를 구성하고있는 요소들을 정의하지만 Schema와는 다르며, 데이터가 정보셋을 가지기 위해서 반드시 유효할(valid) 필요는 없다. XML 데이터가 적법하고(well-formed) XML 구문에 대한 XML 네임스페이스 확장을 따른다면, 그 XML 데이터는 하나의 정보셋을 가진다고 볼 수 있다. XML 정보셋은 XML 문서를 구성하고 있는 정보들을 항목과 프로퍼티로 구분하여 정의한 문서로 W3C에서 XML과 함께 제정하였다. 여기서 항목들은 XML 데이터에 대한 특정 부분의 추상적 표현이다. 또한 XML 데이터의 각 부분은 연관된 프로퍼티 집합을 가진다. 이 용어는 좀 더 일반적인 용어인 트리, 노드, 그리고 프로퍼티와 유사하다. 그러나 항목이나 프로퍼티에서 사용되는 이름들은 XPath나 DCOM 같은 곳에서 다른 데이터 모델과 혼동을 피할 목적에서 사용된다. 정보 항목은 두 XML 관련 스펙(XPath와 DCOM)의 어떤 노드와도 곧바로 연결되지 않는다[4].

2.2 ASN.1

ASN.1은 서로 다른 CPU나 언어 체계에 종속적이지 않은 표준 객체 정의 언어로서 어플리케이션 간의 통신에서 그림 1과 같이 중간자 역할을 하여 데이터 변환을 위한 수고를 덜어주기 위해 고안되었다.

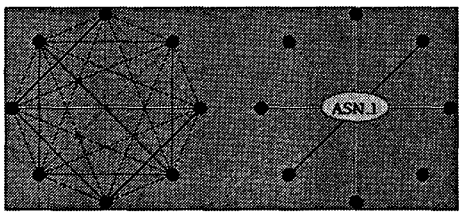

그림 4. ASN.1을 이용한 통신의 형태

ASN.1은 응용 프로그램의 데이터를 추상적으로 정의하거나 응용프로그램의 구조를 정의하기 위해 사용되는 언어로서 BNF(Backus-Naur Form) 형태를 가지는 파스칼이나 C와 비슷한 문법으로 이루어져 있다.

ASN.1은 프로토콜을 정의하는 추상화 구문과 실제 전송 라인에서 어떤 비트 패턴으로 표현될지에 관한 전송 구문을 완전히 분리시켜 놓았다. ASN.1에 의해 표현된 추상적인 프로토콜은 인코딩 룰에 의해 실질적인 비트 패턴으로 변환된다. ASN.1 인코딩 룰은 단대단 간에 통신 및 데이터 전달에서 효율적인 데이터 구조를 지원하기 위한 수단으로 고안된 인코딩/디코딩 방식이며, BER(Binary Encoding Rule), PER(Packed Encoding Rule), XER(XML Encoding Rule) 등 다양한 인코딩 룰이 정의되어 있어 적합한 룰을 가져다 쓸 수 있다. 또한, 고유의 비트 패턴을 정의하고자 한다면 ECN(Encoding Control Notation)을 통해 새로운 룰을 정의할 수도 있다[5].

현재 ASN.1은 ITU-T와 ISO/IEC가 공동으로 표준화를 진행하고 있으며, 각각 ITU-T Study Group 17과 ISO/IEC JTC1 SC6가 표준화를 담당하고 있다. 아래의 표 1은 이러한 ASN.1 표준들을 나열하고 있다.

표 1. ASN.1 관련 표준 목록

규격	제 목
X.680	Specification of Basic Notation
X.681	Information Object Specification
X.682	Constraint Specification
X.683	Parameterization of ASN.1 Specifications
X.690	Specification of Basic Encoding Rules (BER), Canonical Encoding Rules(CER) and Distinguished Encoding Rules(DER)
X.691	Specification of Packed Encoding Rules (PER)
X.692	Specification of Encoding Control Notation (ECN)
X.693	XML encoding rules(XER)
X.694	Mapping W3C XML Schema Definitions into ASN.1

ASN.1은 1988년 X.208(ASN.1)과 X.209(BER)를 시작으로 1994년에 위의 X.691(PER)까지의 표준으로 재구성 되었으며, X.692 이후의 표준은 2000년 이후 추가로 정의되었다.

III. AIGT 시스템의 설계

Fast Infoset 방식을 이용한 바이너리 XML의 인코딩 과정은 크게 두 단계로 나누어 볼 수 있다. 첫 번째 단계는 XML 문서를 정보셋에 따라 파싱하여 ASN.1 value 문서로 재구성하는 과정이다. 이때 파싱

된 XML 문서는 Fast Infoset 방식을 따르도록 구성된 ASN.1 모듈에 따라 정보셋 항목 별로 분류되며 본 문서에는 그 인덱스와 구조표현을 위한 데이터만이 저장되게 된다. 두 번째 단계로는 앞에서 만들어진 ASN.1 문서를 ASN.1 인코딩 룰에 따라 인코딩하여 실제적인 바이너리 문서를 만드는 과정이다. ASN.1에는 BER, CER, DER, PER 등 다양한 인코딩 룰들이 있으며, 본 논문에서는 이 중 가장 압축 효율이 좋은 PER을 사용도록 설계하였다.

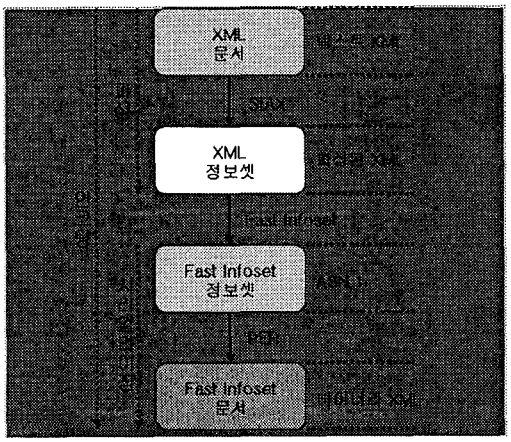

그림 5. Fast Infoset을 이용한 인코딩 흐름도

그림 2은 Fast Infoset 방식을 이용한 XML의 인코딩 흐름도로 파싱된 XML Infoset을 Fast Infoset 모듈로 재구성한 뒤 PER 인코딩을 통해 바이너리화 하는 과정을 보여주고 있다.

3.1 XML 정보셋 처리부

XML 정보셋 처리부는 XML 문서를 파싱하여 XML 정보셋 형태로 변환하는 과정을 담당한다. 그리고 눈에 보이지 않는 파싱된 XML 정보들의 표현을 위해 트리형태의 XML을 보여주고 이를 선택하여 요소들의 정보를 볼 수 있도록 설계하였다. 이러한 XML 문서를 구성하고 있는 요소들의 세세한 정보 표현은 W3C의 XML 정보셋 스펙을 따르도록 하였으며, 파싱 처리를 위해 StAX(Streaming API for XML)를 사용하도록 하였다.

3.2 Fast Infoset 테이블 처리부

ASN.1을 이용해서 바이너리 인코딩을 하기 위해서는 일정한 형식을 유지하는 ASN.1 모듈이 있어야 하기 때문에 XML 문서를 미리 정의된 모듈에 맞추는 재구성 작업이 필요하다. Fast Infoset 테이블 처리부는 XML 문서의 재구성을 위해 Fast Infoset 모듈의테이블 저장소에 파싱된 XML 데이터를 정보셋 항목별로 구분하여 정렬하도록 하였다.

표 2. Fast Infoset 테이블 목록

분류	테이블 명	데이터 형식
단일테이블	· prefix · namespace-name · local-name · other-ncname · other-uri	NonEmptyOctetString : 문자열 형식
	· attribute-value · content-character-chunk · other-string	EncodedCharacterString : 긴 문자열 형식
복합테이블	· element-name · attribute-name	NameSurrogate : prefix, namespace-name, local-name의 인덱스만을 갖는 데이터 형식

ITU-T X.891인 동시에 ISO/IEC 24824-1인 'Information Technology - Generic Applications of ASN.1 : Fast Infoset' 스펙에서는 테이블의 종류를 단일테이블과 복합테이블로 나누며, 각 테이블에 따라 표 2와 같이 데이터 형을 지정하고 있다.

이러한 테이블들을 XML 정보셋 처리부의 XML 트리 요소를 선택하여 어떤 테이블에 몇 번째로 속해 있는지를 보여주도록 하여, XML 정보셋 테이블과 비교가 용이 하도록 설계하였다.

3.3 PER 인코딩 처리부

PER 인코딩은 ASN.1 분야에서 예전부터 사용되어 오던 인코딩 방식으로 XML의 Schema처럼 ASN.1 데이터의 형태를 정의해 주는 언어인 ASN.1 모듈에 의존하여 데이터를 인코딩하는 방식이다. ITU-T Rec. X.691 (2002) ¦ ISO/IEC 8825-2:2002, Specification of Packed Encoding Rules 스펙에서 정의하고 있으며, PER을 사용하면 ASN.1 모듈에 저장되어 있는 요소 외의 실제 내용 데이터만 인코딩되기 때문에 상당한 압축효과를 가져온다[6]. 본 시스템에서는 이러한 PER을 사용하기 위해 Objective Systems, Inc.에서 제공하고 있는 ASN.1 API를 사용하도록 하였다[7].

IV. 시스템의 구현 및 고찰

4.1 구현

본 시스템은 Sun Microsystems에서 진행 중인 프로젝트에 기반한 Fast Infoset 스펙을 따르고 있으며, Windows XP SP2 환경에서 JDK(Java Development Kit) 1.5와 Objective Systems, Inc.의 ASN.1 API를 사용해 구현하였다.

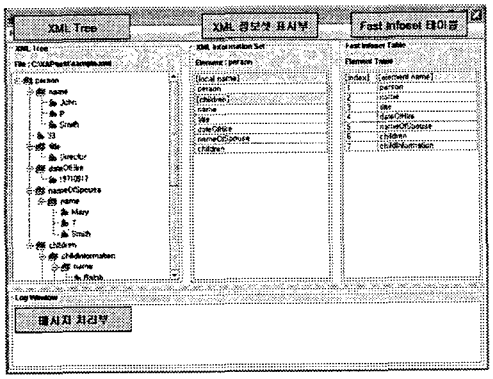

그림 6. XML 바이너리 인코더 화면구성

XML의 파싱은 JAXP(Java API for XML Processing)의 StAX를 사용하였으며, XML·ASN.1·PER로 변환되는 과정을 보여주기 위해 그림 3과 같이 XML 정보셋 표시부, Fast Infoset 테이블부로 UI를 구성하였다.

XML 정보셋 표시부는 XML 문서를 바이너리 인코딩하기 전에 파싱된 XML 데이터의 구성요소와 저장구조를 보여주기 위한 부분으로 XML 문서가 어떠한 항목과 프로퍼티들로 구성되어 있는지 XML 정보셋 스펙에 기반하여 보여주도록 하였다.

Fast Infoset 테이블부는 XML 데이터를 파싱하여 ASN.1 의 Fast Infoset 모듈에 따라 재구성하도록 하였으며, 모듈에 구성된 각각의 테이블에 저장된 데이터를 확인할 수 있도록 테이블 표시창을 구현하였다.

그 외에 상태 표시창은 인코딩이 성공했을 시 작업이 진행되는 과정에서의 소요 시간과 압축률을 표시하며, 실패했을 시 에러 메시지를 보여주도록 하였다. 그리고 XML, ASN.1, PER의 풀 텍스트를 표시하기 위한 메뉴를 구성하였다.

4.2 고찰

본 논문은 XML 문서의 특성을 살리는 Fast Infoset 방식으로 인코딩하여 처리 효율을 떨어뜨리지 않으면서 문서의 크기를 크게 줄일 수 있는 인코더의 설계 및 구현에 관한 것이다.

본 시스템의 특징은 XML 정보셋에 기반한 Fast Infoset 방식을 사용하여 문서의 크기는 줄어들고 처리효율은 떨어지지 않는다는 점이다. 그리고 XML 문서의 인코딩 과정을 눈으로 확인 할 수 있도록 결과 문서로는 나타나지 않는 XML 구성요소와 Fast Infoset 모듈 구성요소를 보여주도록 하여 인코딩 과정을 알기 쉽게 구현하였다.

그러나 Fast Infoset 방식만이 XML 바이너리 인코딩에 관한 표준이 아니며, Fast Infoset 방식보다 효율이 뛰어난 Fast Schema 방식이나 둘을 병용하는 하이브리드 방식 등이 속속 등장하고 있는 추세이다. 이에 최적화된 XML 바이너리 인코딩을 위해서 최신 방식들에 대한 적용이 고려되어야 할 것이다.

V. 결 론

XML의 사용이 점차 보편화 되어가고 있는 가운데 모바일이나 그리드 컴퓨팅과 같이 파일의 크기가 시스템의 성능에 큰 영향을 주는 분야의 경우 XML 문서의 크기를 줄일 수 있는 바이너리 인코딩에 관심이 높아지고 있다.

썬 마이크로시스템즈에서는 이에 대한 방안으로 Fast Infoset 프로젝트를 진행 중에 있으며, XML 정보셋 스펙과 ASN.1을 이용한 인코딩 방식을 제안하여 표준화를 진행하고 있다.

이에 본 논문에서는 XML 문서가 어떻게 Fast Infoset 방식으로 인코딩 되는 과정을 보여 줄 수 있도록 시스템을 구현하였다. XML 문서가 가지는 요소들을 처리 전 형태로 XML 정보셋에 따라 보여주었으며, 처리 후에는 Fast Infoset 테이블에 저장된 요소들을 보여주도록 하여 변환된 문서구조를 알기 쉽도록 처리하였다.

향후 연구 과제로는 현재 나온 Fast Inforset 뿐만 아니라 XML Schema를 이용한 Fast Schema 방식을 통한 바이너리 인코딩 기능을 추가해야 할 것이다. 또한, Web Services에 바이너리 XML을 적용한 Fast Web Services 분야에 대한 연구도 필요할 것이다.

참고문헌

[1] W3C, Extensible Markup Language, http://www.w3.org/XML/

[2] Fast Infoset, http://java.sun.com/developer/technicalArticles/xml/fastinfoset/

[3] ITU-T Rec. X.891(2005) | ISO/IEC 24824-1, Fast Infoset

[4] XML Information Set (Second Edition), http://www.w3.org/TR/xml-infoset/

[5] ASN.1 Information Site, http://asn1.elibel.tm.fr/

[6] ITU-T Rec. X.691 (2002) | ISO/IEC 8825-2:2002, Specification of Packed Encoding Rules (PER)

[7] Objective system, Inc., http://www.obj-sys.com/docs/acv58/JavaRunTime/