
Fast 웹서비스를 위한 Fast XML 인코딩 시스템

김종문 · 우뢰 · 홍현우 · 최봉규 · 정희경

배재대학교 컴퓨터공학과

A Fast XML Encoding System for Fast Web Services

Jong-Moon Kim · Lei Yu · Xian-Yu hong · Bong-Kyu Choi · Hoe-Kyung Jung

Dept. of Computer Engineering, Paichai University

E-mail : {elcomtech, yulei0622, hongxianyu, bongkyu1963choi, hkjung}@pcu.ac.kr

요 약

웹서비스는 운영환경에 독립적인 XML(Extensible Markup Language)을 사용하여 서로 다른 플랫폼간의 통합을 가능하게 하고 있다. 그러나 XML은 불필요한 텍스트 데이터를 많이 가지고 있기 때문에 무선 인터넷이나 이동통신모뎀과 같이 상대적으로 느린 통신 매체에 자주 접속하거나 자원이 한정적인 소형 기기에서 사용할 경우 전체적인 성능을 저하시키는 문제점이 있다. 이에 XML을 바이너리화 하여 문서의 크기를 줄이는 표준을 ITU-T(International Telecommunication Union - Telecommunication)와 ISO(International Organization for Standardization)/ IEC(International Electrotechnical Commission)에서 공동으로 제안하게 되었으며, 현재 진행 중인 바이너리 XML 인코딩 표준으로는 Fast Infoset과 Fast Schema가 있다.

이에 본 논문에서는 웹서비스의 성능 향상을 위해 현재 표준화가 진행 중인 Fast Infoset 알고리즘과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 구현하였다.

ABSTRACT

Web service in operating environment independent XML about under using. It dose so, Integration of the platform for it is different each other possibly it is doing. But XML in order to have the text data which is unnecessary plentifully with wireless Internet or the mobile communication modem in together relatively will connect frequently slow communication medium and there is a problem point which decreases the case prerequisite efficiency which the resources will use from the limited small-sized machinery and tools. Hereupon XML about under make binary the standard which reduces the size of the document from ITU-T and ISO/IEC it was under proposing with commonness, currently binary XML encoding where it is in the process of advancing standard there are Fast Infoset and Fast Schema.

In this paper, implementation of Fast XML encoding system through introduction Fast Infoset algorithm and Fast Schema algorithm for web services increase performance.

키워드

XML, Web Services, Fast Web Services, Fast Infoset, Fast Schema

제1장 서 론

서로 다른 환경에서 사용되는 어플리케이션들은 플랫폼과 프로그래밍 언어에 종속적이기 때문에 상호운용성이 결여되어 웹의 발전에 걸림돌로 작용하고 있다. 이에 웹에서의 상호운용성 확보를 위해 운영환경에 독립적인 문서 형식을 정의할 수 있는 XML(Extensible Markup Language)을

기반으로 한 웹서비스(Web Services)가 등장하게 되었다.

웹서비스는 서로 다른 플랫폼간의 통합을 가능하게 하였으나, XML은 불필요한 텍스트 데이터를 많이 가지고 있어 무선 인터넷이나 이동통신모뎀과 같이 상대적으로 느린 통신 매체에 자주 접속하거나 자원이 한정적인 소형 기기에서 사용시 전체적인 성능을 저하시키는 문제점이 있다.

현재까지 자원이 한정적인 환경에서의 XML 성능 문제는 Gzip과 같은 단순 압축 방식을 이용해 해결하고자 하였으나, XML에 적합하지 않은 압축 방식들은 XML 문서의 크기만 줄일 뿐 근본적인 성능 개선이 안 되어 ITU-T와 ISO/IEC에서 공동으로 바이너리 XML 인코딩 표준을 제안하게 되었으며, 바이너리 XML 인코딩 표준으로는 Fast Infoset과 Fast Schema가 있다[1,2].

본 논문에서는 웹서비스의 성능 향상을 위해 현재 표준화가 진행 중인 Fast Infoset 알고리즘과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 구현하였다.

제2장 관련연구

2.1 Fast 웹서비스

XML 기반 메시지는 현재 웹서비스 기술의 중심이고, SOAP(Simple Object Access Protocol) 메시지는 기존 프로토콜보다 큰 메시지를 가지며, 많은 계산을 필요로 한다. 이는 SOAP 메시지의 데이터 처리가 비효율적이기에 웹서비스를 보다 빠르게 처리하기 위해 XML의 바이너리 인코딩이 필요하게 되었으며, 바이너리 인코딩 기반의 Fast 웹서비스 시스템 구축이 요구되고 있다.

Fast 웹서비스의 요점은 웹서비스에서 사용되는 XML 데이터의 크기를 줄여 전송과 처리에 소요되는 시간을 줄이는 것이다. Fast XML은 문서의 크기는 줄어들면서 추가적인 압축과 해제 과정이 필요하지 않으며, XML 문서와 마찬가지로 파싱(Parsing) 및 시리얼라이징(Serializing)이 가능하다. 따라서 기존 웹서비스의 단점을 보완하고자 Fast 웹서비스에서는 Fast XML로 인코딩된 SOAP 메시지를 사용한다[3].

2.2 Fast Infoset

Fast Infoset은 XML Infoset에 기반한 인코딩 표준으로서 테이블과 인덱스를 사용하여 XML 데이터를 인코딩 한다[4]. Fast Infoset 알고리즘은 XML 정보셋에 따라 구분된 정보들을 테이블에 저장하여 인덱싱(Indexing)한 후, XML 문서의 문자들을 인덱스로 대체하여 문서의 크기를 줄인다.

Fast Infoset 알고리즘을 이용한 바이너리 인코딩은 XML 문서를 파싱하여 XML 정보셋으로 변환하고, Fast Infoset 알고리즘에 따라 각 정보셋을 어휘(Vocabulary) 테이블에 저장하고 인덱스로 대체하여 Fast Infoset 데이터로 재구성한다.

2.3 Fast Schema

Fast Schema는 XML 문서의 Character 정보 항목을 제외한 모든 정보 항목을 Schema 문서로 처리하고 Character 정보 항목만을 인코딩하여 문서의 크기를 줄이는 방식이다. Fast Schema 알고리즘은 다음과 같은 특징을 가진다[5].

첫째, 높은 압축률로 Fast Schema 알고리즘은 XML 문서의 정보셋과 구조가 모두 Schema 문서에 포함되어 있어, Schema 문서의 특성상 미리 정의된 문서는 참조가 간편하고, 따로 전송할 필요가 없어 XML 문서의 압축에 효과적이다.

둘째, 구조 데이터의 최소화로 Schema 문서에는 Element 이름 등과 같이 대부분의 정보가 표현되어 있기 때문에 Fast Schema 문서에서 사용되는 구조 데이터는 몇몇 특수한 경우의 숫자 표현과 문자열의 길이 표현만으로 이루어져 있다.

셋째, 사용 제한으로 Fast Schema 알고리즘은 Schema 문서를 사용해 문서 구조를 처리하므로 Schema 구조가 미리 정의되어 있는 XML 문서만이 사용 가능하다.

제3장 인코딩 시스템 설계

Fast XML 인코딩 시스템은 사용되는 알고리즘에 따라 Fast Infoset 변환부와 Fast Schema 변환부로 분류하였으며, 각각의 알고리즘 처리를 위한 데이터를 제공하는 XML 파싱부를 두어 처리 과정을 통합하여 구성하였다. 그림 1은 Fast XML 인코딩 시스템 전체 구성도이다.

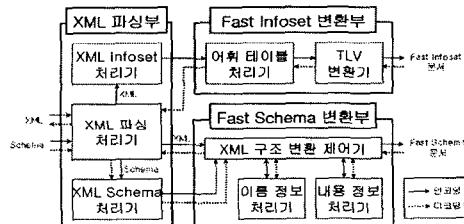


그림 4. 인코더 전체 구성도

3.1 XML 파싱부

XML 파싱 처리기는 파서를 통한 XML 문서의 파싱 및 시리얼라이징 부분을 담당한다. 인코딩은 Fast Infoset 알고리즘과 Fast Schema 알고리즘에 맞는 데이터를 선별하는 XML Infoset 처리기와 XML Schema 처리기에 파싱된 XML 및 Schema 문서를 전달하며 디코딩은 Fast Infoset 변환부와 Fast Schema 변환부에서 파싱된 XML 데이터를 받아 XML 문서로 시리얼라이징 한다.

XML Infoset 처리기는 XML Infoset 데이터를 분류하여 Fast Infoset 변환부의 어휘 테이블 처리기와 TLV(Tag, Length, Value) 변환기에서 사용할 수 있도록 제공한다.

XML Schema 처리기는 파싱된 Schema 문서에서 구조 정보를 추출하여 Fast Schema 변환부의 XML 문서 구조 제어기에 전달한다. XML Schema 처리기는 직접 변환되는 데이터가 아닌 변환을 돋는 Schema 데이터의 처리를 담당하므로 인코딩, 디코딩 시 같은 역할을 수행한다.

3.2 Fast Infoset 변환부

Fast Infoset 변환부는 XML Infoset 데이터를 테이블에 저장하여 인덱스를 부여하는 어휘 테이블 처리기와 인덱싱 처리된 XML Infoset 데이터를 TLV 형식으로 바이너리 인코딩하여 Fast Infoset 문서를 만드는 TLV 변환기로 구성된다.

어휘 테이블 처리기는 Fast Infoset에 정의된 10가지의 어휘 테이블을 관리하는 모듈로서 XML 문서의 중복되는 정보 항목들을 테이블에 저장하여 인덱스로 대체하는 역할을 담당한다.

Fast Infoset 알고리즘을 통해 변환된 XML 정보 항목은 태그, 길이, 내용의 순서로 구조화 된다. XML Infoset의 변환 흐름은 XML 문서의 파싱 흐름과 일치하므로 XML 문서 구조를 따른다.

3.3 Fast Schema 변환부

Fast Schema 변환부는 파싱된 XML 데이터를 받아들여 문서 변환 처리하는 XML 구조 변환 제어기를 중심으로 이름 정보 처리기와 내용 정보 처리기가 이를 정보와 내용 정보 처리를 담당하는 형태로 구성된다.

XML 구조 변환 제어기는 Fast Schema 변환부의 중심으로 이름 정보 처리기와 내용 정보 처리기를 호출하여 변환을 제어하며 그림 2는 XML 구조 변환 제어기의 SOAP문서 변환 흐름도이다.

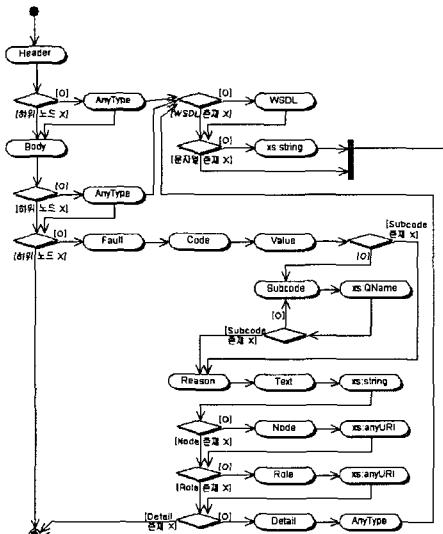


그림 5. SOAP 문서 변환 흐름도

이름 정보 처리기의 역할은 크게 이름 정보의 제거와 이름 정보를 대신할 구조 데이터의 생성으로 나눌 수 있다. XML 문서의 제공자와 요청자가 모두 Schema 문서를 가지고 있다면 XML의 데이터 구조를 표현하는 Element와 Attribute의 이름 정보는 필요 없으며, 이름 정보를 제거함으로써 직접적인 내용 정보만을 전송할 수 있게 된

다. 그러나 XML 문서에는 Schema 문서만으로는 알 수 없는 구조 정보가 있으며, 이 경우 Fast Schema 문서에 구조 데이터를 삽입하여 문서 구조를 유지한다.

내용 정보 처리기는 Character, Attribute Value 정보 항목 등 문서에서 담고자 하는 내용 정보를 처리한다. Fast Schema 알고리즘을 통해 변환된 XML 문서는 태그가 지워지고 Element, Attribute 이름이 제거되었으므로, 내용 정보의 바로 앞에 길이 정보를 붙여 내용의 시작과 끝을 알 수 있도록 하였다.

제4장 인코딩 시스템의 구현

구현 환경은 Windows XP SP2 환경에서 JDK(Java Development Kit) 1.5를 사용하여 Java로 구현하였다. 또한 Fast XML 인코딩 알고리즘들의 성능 분석 및 테스트를 위해 구현되었으며, XML 문서, Fast Infoset 문서, Fast Schema 문서들의 구조 및 내용 파악이 용이하도록 구현하는데 초점을 맞추었다.

4.1 메인 화면 구성

문서 구조를 파악하기 위한 XML Tree를 화면 왼쪽에 배치하였으며, XML 문서, Fast Infoset 문서, Fast Schema 문서 파일을 보고 저장할 수 있는 버튼들과 인코딩 성능을 비교하는 결과표 부분, 그리고 변환 메시지를 보여주는 메시지 표시 부분으로 구성되도록 그림 3과같이 구현하였다.

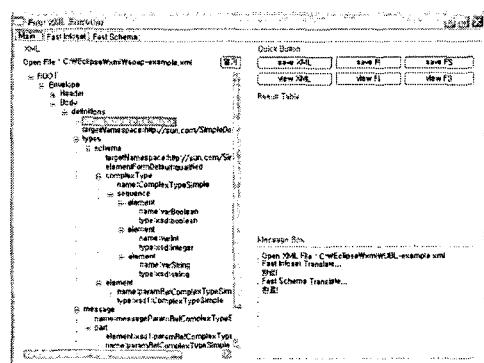


그림 6. 메인 인터페이스

4.2 Fast Infoset 화면 구성

Fast XML 인코더의 Fast Infoset 탭에서는 인코딩 된 "finf" 파일을 저장할 수 있으며, 트리를 통해 정보 항목을 선택하면 Fast Infoset 테이블 부분은 변환된 Fast Infoset 문서의 바이너리 데이터를 비트 단위로 잘라서 정보 항목과 비교 설명하고 어휘 테이블 부분에는 해당 정보 항목의 상세 정보를 보여주도록 그림 4와 같이 구현하였다.

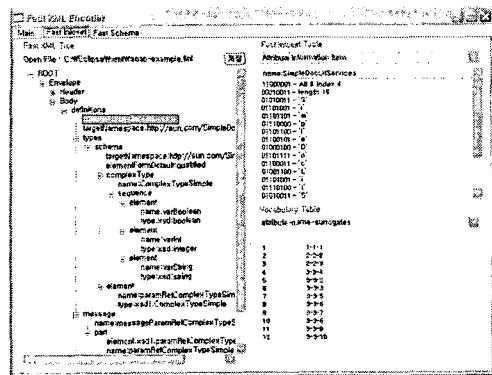


그림 7. Fast Infoset 인터페이스

4.3 Fast Schema 화면 구성

Fast XML 인코더의 Fast Schema 탭에서는 인코딩된 "mfs" 파일을 저장할 수 있으며, XML Schema 태이블 부분은 변환된 Fast Schema 문서의 바이너리 데이터를 비트 단위로 정보 항목과 비교 설명하고 Schema 부분은 Schema 문서에서 선택된 정보 항목의 구조를 정의하고 있는 타입 부분을 텍스트로 나타내도록 그림 5와 같이 구현하였다.

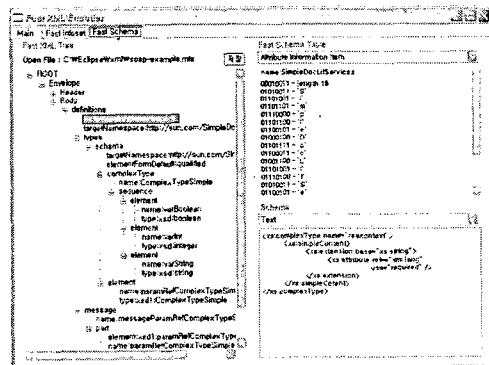


그림 8. Fast Schema 인터페이스

제5장 결과 분석 및 고찰

본 논문에서는 신뢰할 수 있는 성능 평가를 위해 문서 크기와 정보 항목의 수를 고려하여 총 10개의 SOAP 문서를 테스트 하였다.

Fast XML 인코딩 테스트에서는 Fast 웹서비스를 위해 SOAP 문서만을 사용하므로 정보 항목의 종류는 거의 일정하다. 이는 SOAP 문서의 정보 항목 수가 많아서 문서의 크기가 커지면, 정보 항목의 중복이 늘어나게 된다는 것을 의미한다. 그림 6은 테스트 결과를 비교 분석한 그래프이다.

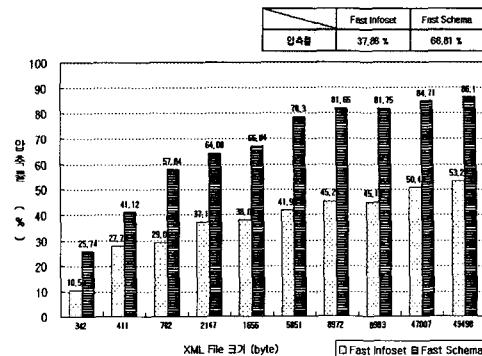


그림 9. 테스트 결과 그래프

테스트 결과를 보면 Fast 웹서비스 구축시 필요한 Fast XML 인코딩이 SOAP Schema에 의존적이므로 모든 XML 문서를 인코딩 할 수 있는 Fast Infoset 방식보다는 SOAP Schema에 국한시킨 Fast Schema 방식이 뛰어난 것으로 판단된다.

제6장 결 론

모바일이나 대용량 서버 등 모든 플랫폼의 어플리케이션을 통합하는 진정한 서비스 통합을 이루기 위해서는 웹서비스의 성능을 개선한 Fast 웹서비스의 구축이 필요하다. 그러나 사용되는 메시지인 XML 문서는 표현하고자 하는 부분의 자료가 장황하기 때문에 부하가 큰 문제점을 가지며, 다양한 플랫폼 환경에서 Fast 웹서비스 구축시 상호운용성이 확보되어야 하나 개발자들이 참조할 수 있는 가이드라인이 없어 큰 혼란이 예상된다.

이에 본 논문에서는 기존의 웹서비스에서 사용되던 XML 문서의 과부하 문제를 해결하기 위해 Fast Infoset과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 설계 및 구현하였다.

향후 연구과제로는 현재 구현된 Fast Schema 알고리즘을 SOAP과 WSDL 뿐만 아니라 Schema를 가지는 모든 XML 문서를 인코딩 할 수 있도록 확장시킬 필요가 있다.

참고문헌

- [1] 최일선 배재대학교 박사논문, “Fast Infoset 기반 Fast 웹서비스에 관한 연구”, 2004.
- [2] 이강찬, “웹서비스 기술 표준화 및 도입 이슈”, 2003.
- [3] XML Binary Characterization Working Group , <http://www.w3.org/XML/Binary/>
- [4] ITU-T Rec. X.891 (2005) | ISO/IEC 24824-1, Fast Infoset
- [5] ITU-T Rec. X.892 (2005) | ISO/IEC 24824-2, Fast Web Services