

---

# Fast 웹 서비스를 위한 Fast Infoset 알고리즘 구현

조태범\* · 박연식\*\* · 정희경\*

## Implementation of Fast Infoset Algorithm for Fast Web Services

Tae-Beom Cho\* · Yeoun-Sik Park\*\* · Hoe-Kyung Jung\*

### 요 약

기존의 웹 서비스(Web Services)는 운영환경에 독립적인 문서 형식을 정의할 수 있는 XML(Extensible Markup Language)을 기본 문서 형식으로 사용함으로써 서비스의 통합을 실현하였다. 그러나 텍스트 데이터로 이루어져 있는 XML 문서는 네트워크 환경이나 임베디드 시스템과 같이 상대적으로 느린 통신 매체에 자주 접속하거나 모바일과 같이 자원이 한정된 소형 기기에서 사용 할 경우 전체 응용 프로그램의 성능을 저하시키는 요소로 작용한다.

이에 본 논문은 Fast Infoset 방식과 ASN.1의 PER 인코딩 기법을 사용하여 XML 문서를 Fast XML으로 변환 할 수 있는 Fast Infoset 알고리즘을 구현하였고, 테스트 베드(Test Bed)를 구축하였다. 또한, 웹 서비스 구축 시 전송 프로토콜로 사용되는 XML 기반의 SOAP 메시지를 인코딩하여 기존의 순수 XML 메시지 방식과의 처리 성능 차이를 비교 분석하였다.

### ABSTRACT

Plain old Web Services came true a services integration with use the XML that will be able to define a platform-independent document format in basic document format. But, the XML document decreases the efficiency of the whole application program with connects frequently in relatively slow communication media like network environment, embedded system or use the resource limited small-sized instrument like mobile.

In this paper, we implemented the algorism of Fast Infoset, which can convert XML into Fast XML using the Fast Infoset method and PER encoding rules in ASN.1 and constructs Test Bed. Also, it is compared with the existing pure XML message method in difference of process performance after encoding SOAP messages of XML basis when constructing web service.

### 키워드

Web Services, Infoset, XML, ASN.1

## 1. 서 론

현재까지의 웹 서비스 시스템은 효과적으로 서비스를 지원하고, 다양한 플랫폼 환경을 통합 연동하는 방향으로 연구가 이루어졌다. 그러나 적은 대역폭이나 자원

이 제한된 휴대용 기기 같은 특화 분야에서는 XML 메시지 크기로 인해 웹 서비스를 사용하는데 제한을 가지고 있다[1]. 이에 웹 서비스 성능 향상을 위해 대역폭의 증설이나 휴대 기기 자원의 확장 방법을 사용하지 않고 인코딩(encoding) 방법을 사용하여 웹 서비스의 성능을 향

---

\* 배재대학교 컴퓨터공학과 (교신저자 : 정희경)

\*\* 경상대학교 교수, 해양산업연구소 연구원

상시키는 바이너리(binary) 인코딩 방안이 대두 되었다.

XML 문서에 대하여 바이너리 인코딩 방식의 특징은 전송량이나 처리 성능의 효율 면에서 상당히 우수하다. 바이너리 인코딩 방식을 사용하지 않고 다른 방식을 사용해 XML 문서를 압축하는 것은 처리 성능을 향상시키는데 효과적 일 수 있지만, XML의 호환성이 보장되지 않을 때는 심각한 결과를 초래할 수 있다[1,2].

그러나 이러한 호환성을 제한하는 문제에도 불구하고 XML 문서를 압축 방식으로 처리하여 XML 문서의 크기를 줄이려는 것은 특화 분야에서 전송량이나 처리 성능 개선의 폭이 크기 때문이다.

이에 본 논문에서는 XML의 호환성을 보장하기 위해 바이너리 인코딩 표준인 Fast Infoset 알고리즘을 도입하였으며, Fast Infoset 인코딩 시스템을 구현하여 Fast 웹 서비스 아키텍처를 제시한다. 또한, 시뮬레이션과 분석 등의 검증 작업을 통하여 Fast 웹 서비스 개발자나 사용자들이 Fast 웹 서비스 시스템을 구축하는데 있어서 참고할 수 있는 테스트 베드를 제공하고자 한다.

## II. 관련 연구

Fast 웹 서비스는 기존 웹 서비스에서 사용되던 XML 메시지를 XML 정보셋과 ASN.1(Abstract Syntax Notation One)을 이용해 바이너리 인코딩 한다.

웹 서비스에서 서비스 제공은 XML 메시지를 사용하고 있다. XML은 다양한 정보 항목(Information Items)으로 구성되어 있으며, XML 정보셋으로 정의된다. ASN.1은 XML의 효율적인 정보 처리를 위해 XML 문서를 바이너리 형태로 인코딩하며 Fast 웹 서비스를 제공할 수 있도록 사용된다.

### 2.1. XML 정보셋(Information Set=Infoset)

XML 데이터는 하나의 정보셋을 이룰 수 있는데, 정보셋은 11가지 정보 항목으로 구성된다. 정보 항목들은 XML 데이터에 대한 특정 부분의 추상적 표현으로 XML 데이터의 각 부분에 연관된 프로퍼티(property)를 가진다.

프로퍼티는 트리나 노드와 유사하지만 XPath나 DOM(Document Object Model)과 같은 곳에서 사용된 다른 데이터 모델과의 혼동을 피할 목적으로 사용한다. 이

러한 이유 때문에 정보 항목은 XML 관련 표준의 어떤 노드와도 곧바로 연결되지 않도록 정의되어 있다[3,4].

XML 정보셋은 추상적인 개념이기 때문에, 이를 구현하기 위한 여러 방법이 있다. 정보셋을 간단하게 계층적 트리로 처리하였지만, 반드시 계층적 트리 형태로 표현할 필요는 없으며 SAX(Simple API for XML)와 같은 이벤트 기반 인터페이스나 질의 기반 인터페이스로도 표현할 수 있다.

### 2.2. Fast Infoset

Fast Infoset 표준 명세서는 ASN.1의 일반적인 어플리케이션으로 분류되며, 바이너리 인코딩은 ASN.1 표기법과 ECN(Encoding Control Notation)을 사용해 XML 정보셋을 바이너리 형식으로 변환하는 방법을 기술하고 있다[5,6]. 또한 인코딩 방식을 사용하여 XML 메시지 크기를 최대한 압축하여 Fast Infoset 문서의 생성과 처리 속도를 높일 수 있는 기술을 나타낸다. Fast Infoset 표준 명세서에는 어휘 테이블 생성과 인덱스의 사용, Fast Infoset 처리를 위한 ASN.1 타입 정의, 제한된 알파벳 사용, 인코딩 알고리즘 등에 관한 상세한 내용이 기술되어 있다[6].

이 표준 명세서의 처리 과정은 XML 문서의 파싱 과정으로 XML 정보셋을 정보 항목으로 분류하고, 파싱된 XML 정보셋은 Fast Infoset의 어휘 테이블과 인덱스 규칙에 따라 Fast Infoset value 형태로 재구성되며 ASN.1 인코딩 규칙을 통해 최종적으로 바이너리 형식의 문서로 변환된다.

단점으로는 Fast Infoset 문서를 개발자나 사용자가 보는데 바이너리 내용이 포함되어 문서의 내용을 쉽게 알기가 힘들다는 점이다.

### 2.3. ASN.1

ASN.1은 추상구문을 기술하기 위한 추상 구문 표기법을 정의하고, 그 추상구문을 전송 구문으로 변환하는 규칙을 규정하고 있다. 응용 프로그램 데이터를 추상적으로 정의하거나 응용 프로그램의 구조 정의, PDU(Protocol Description Unit)들을 정의하는데 사용되며, 추상 문법으로 기본적인 데이터의 구조를 기술하고 새로운 타입과 그에 해당될 수 있는 값을 정의하는 방법이다 [5].

데이터 타입은 임의 값들의 공통된 이름으로 단순히

거나 여러 가지 타입이 복합될 수도 있다. 인코딩은 데이터 값을 나타내는 옥텟(octet)으로 배열하며, 인코딩 규칙은 한 구문을 다른 구문으로 바꾸어 주는 규칙이고, 전송 문법은 데이터가 실제 전송될 때 나타나는 비트 형태를 기술하는 문법을 사용한다[7,8].

### 2.4. 기존 웹 서비스의 인코딩 문제

웹 서비스에서 사용되고 있는 XML 기반 메시지는 현재 웹 서비스 기술의 중심에 있다. XML 기반 메시지는 기존 프로토콜 RMI(Remote Method Invocation), RMI/IIOP (Internet Inter-ORB Protocol) 또는 CORBA/IIOP와 같은 기존 프로토콜보다 더 많은 처리량과 더 큰 메시지를 가지며, 더 많은 계산을 필요로 한다. 이는 기존의 프로토콜 방식보다 XML 기반 메시지의 데이터 처리가 비효율적인 것을 의미한다. 또한, 더 큰 메시지를 전송할 경우 작은 메시지로 나누어 재전송하는 경우가 많아 장시간의 네트워크를 점유하는 결과를 초래하고 있다. 무선의 낮은 대역폭은 종종 이것으로 인해 오류가 발생하는 경우도 있다[1,4,9].

따라서, 웹 서비스를 보다 빠르게 처리하기 위해 인코딩 방식으로 구현된 Fast Infoset을 사용하고, 이를 이용한 웹 서비스 시스템 구축을 통해 보다 빠른 웹 서비스를 할 수 있는 환경이 요구되고 있다.

그러나 현재 웹 서비스 개발자를 위한 가이드라인이 제공되어 있지 않기 때문에 향후 Fast 웹 서비스 시스템을 개발하는데 있어 많은 시행착오와 혼란이 가중될 것으로 보인다. 현재 웹 서비스의 단점을 보완하기 위해 XML 문서를 Gzip 등으로 압축하여 사용하고 있으나 압축된 문서를 해제하는 과정에서 처리 효율이 떨어지는 문제가 있다.

### III. Fast 웹 서비스를 위한 아키텍처 설계

Fast 웹 서비스의 요점은 웹 서비스에서 사용되는 XML 데이터의 크기를 줄여 전송 속도와 처리 성능의 효율을 향상시키는 것이다. 이를 위해서는 웹 서비스의 전송 프로토콜인 SOAP(Simple Object Access Protocol) 메시지와 전송되는 서비스를 기술하는 언어인 WSDL (Web Service Definition Language)에 대한 바이너리화가 중요하다.

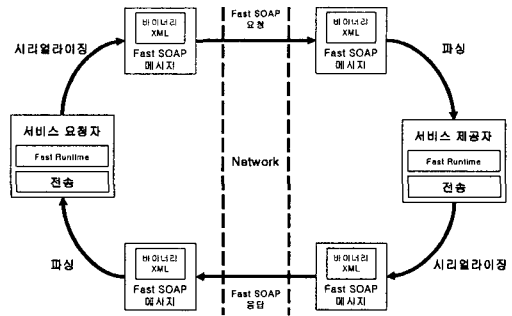


그림 1. Fast 웹 서비스 구조  
Fig. 1. Structure of Fast Web Services

그림 1은 Fast 웹 서비스의 XML 메시지 전송과 처리에 관한 대략적인 구조를 나타내고 있다. 기존의 웹 서비스에서는 서버와 클라이언트 간에 XML이나 압축된 XML로 메시지를 주고받았으나 XML 메시지는 전송량이 크다는 단점이 있고, 압축된 XML 메시지는 전송 시 압축과 해제 과정이 추가되어 처리시간이 늘어난다는 단점이 있다. 따라서 기존 웹 서비스의 단점들을 보완하고자 Fast 웹 서비스에서는 SOAP과 WSDL 대신 Fast XML로 인코딩된 Fast SOAP을 사용한다.

### 3.1. Fast 웹 서비스를 위한 인코딩

그림 2는 XML 문서가 Fast Infoset 알고리즘을 통해 바이너리화 되는 과정을 보여주고 있다. 우선 XML 문서에서 정보 항목들을 분류해 내기 위한 파싱 과정을 거친 후 Fast Infoset 알고리즘에 따라 문서를 재구성한다. 그리고 ASN.1의 인코딩 규칙 중 하나인 PER을 이용하여 문서를 최종적으로 바이너리화 하게 된다.

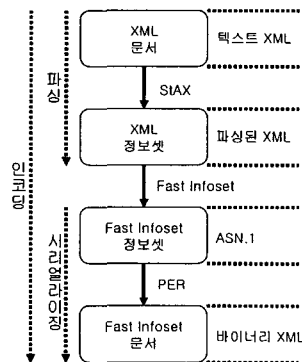


그림 2. Fast Infoset 흐름도  
Fig. 2. Activity diagram of Fast Infoset

### 3.2. Fast Infoset에 기반한 인코더 설계

XML 문서를 Fast Infoset 문서 형태로 인코딩하는 과정은 XML 문서를 Fast Infoset 명세에 정의된 Fast Infoset value 형태의 ASN.1 값으로 변환하는 과정과 이를 ASN.1 인코딩 규칙을 통해 바이너리화 과정으로 나누어진다[8,10].

본 논문에서는 인코딩을 위해 Fast Infoset 명세에 정의된 ASN.1 모듈에 따라 XML의 재구성 과정을 설계하였으며, ASN.1 모듈을 인코딩하기 위한 ASN.1 인코딩 규칙으로 압축 효율이 뛰어난 PER을 사용하였다[10,11,12].

그림 3에서 XML 정보셋의 11가지 정보 항목을 보여주고 있으며, Fast Infoset 알고리즘을 통한 XML 문서의 변환 흐름을 나타낸다.

Document 정보 항목이 정의하고 있는 Fast Infoset의 어휘 테이블은 정보 항목의 식별을 위해 사용하는 문자열 테이블, Character 정보 항목이나 Attribute 정보 항목의 값과 같이 문서의 내용을 담고 있는 긴 문자열 테이블, Element와 Attribute의 이름 프로퍼티를 효율적으로 나타내기 위한 복합 테이블로 구성된다.

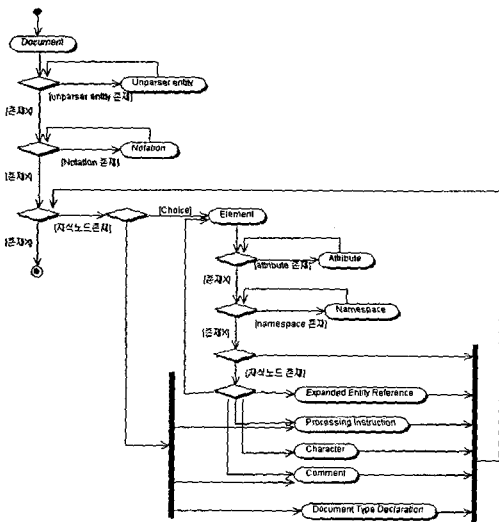


그림 3. XML 정보셋 항목 전체 처리 흐름도  
Fig. 3. Activity diagram of XML Infoset

ASN.1의 인코딩 규칙을 이용한 바이너리 인코딩 과정에서는 ASN.1 인코딩 규칙 중 데이터 압축 성능이 가장 뛰어난 PER을 사용하여 변환한다[6,11,12].

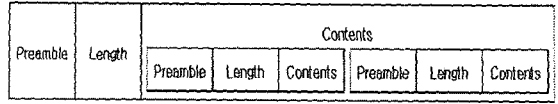


그림 4. PER 인코딩 구조  
Fig. 4. Structure of PER encoding

그림 4는 PER 인코딩의 기본적인 구조를 보여주고 있다. PER 방식은 문서 구조를 표현하기 위한 자식요소의 몇 가지 정보를 표현하는 Preamble, Contents의 총 길이를 표현하는 Length, 자식 요소로 나타나는 Element, Attribute, PI, Character Chunk, Attribute Value 등의 데이터를 인덱스 또는 문자열로 표현하는 Contents의 구조로 데이터를 표현한다[11].

## IV. 시뮬레이션

시뮬레이션 환경은 Windows XP SP2 환경에서 JDK(Java Development Kit) 1.5를 사용하여 구현하였다. XML 문서의 파싱을 위한 API로는 JAXP(Java API for XML Processing)의 StAX를 사용하였다[13,14,15].

그림 5는 XML 문서를 Fast Infoset 방식에 따라 인코딩하는 과정을 보여주기 위한 사용자 인터페이스 화면이다.

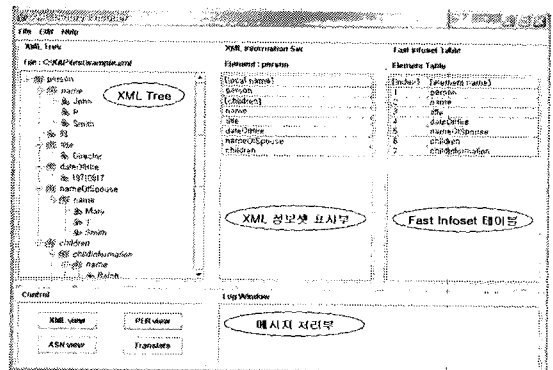


그림 5. 사용자 인터페이스  
Fig. 5. User interface

시스템은 파싱된 XML 노드들을 Fast Infoset 규칙에 따라 ASN.1 value로 재구성하기 위해 ASN.1 모듈에 따라 데이터 저장공간을 구조화한 FastInfosetValue.java와 FastInfosetValue 인스턴스를 사용하여 XML 노드를 처

리하는 FastInfosetRemodeling.java, 그리고 테스트를 위한 사용자 인터페이스를 구현한 XAPEncoder 와 TextView 클래스로 구성되어 있다.

테스트를 위한 예로는 도서에 대한 정보를 담고 있는 간단한 문서를 사용하였다. 그림 6은 원본 XML 문서를 텍스트 형태로 나타내는 TextView를 보여준다. 그림 7은 Book.xml 데이터를 Fast Infoset 형태로 변환하여 보여 주고 있으며, Fast Infoset의 어휘 테이블들이 어떻게 XML 데이터를 저장하는지를 보여준다. 그림 8은 이러한 ASN.1 value의 데이터를 구조화하여 표현한 것이다.

ASN.1 value는 실제적인 문서로 볼 수 없는 변환 과정 중의 데이터이며, ASN.1 value를 PER 인코딩 하여 Fast XML 문서를 얻는다.

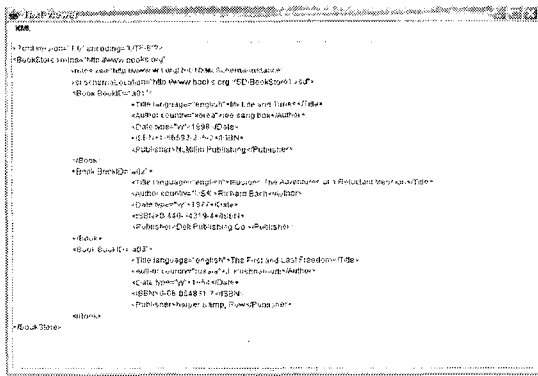


그림 6. Book.xml 문서  
Fig. 6. Book.xml document

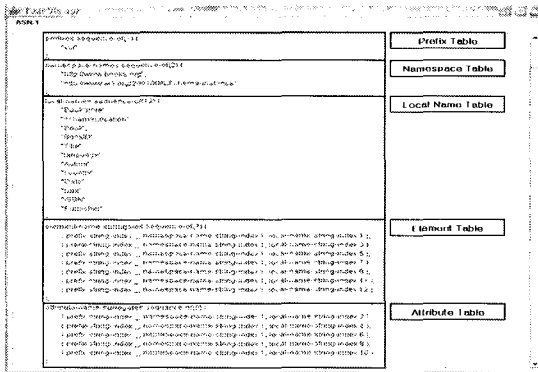


그림 7. Fast Infoset 어휘 테이블  
Fig. 7. Vocabulary Table of Fast Infoset

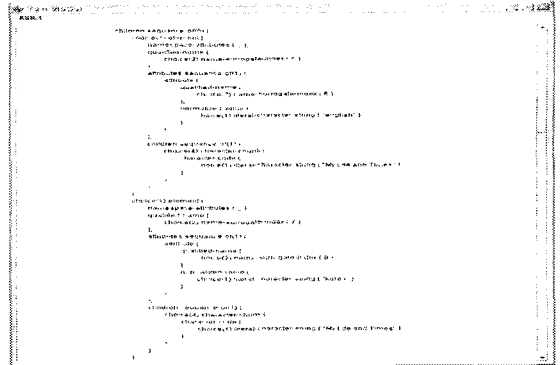


그림 8. Fast Infoset 처리 결과  
Fig. 8. Processing result of Fast Infoset

그림 9는 Fast 웹 서비스를 위한 테스트 시나리오를 나타낸 것으로 웹 서비스 요청자가 먼저 웹 서비스 제공자에게 원하는 정보를 XML로 보내게 되면, 웹 서비스 요청자의 중간에 위치한 Fast Infoset 모듈에서 XML 문서를 파싱한 후 테이블과 인덱싱을 통해 Fast Infoset value 형태의 ASN.1 value로 변환한다. ASN.1 value 값은 ASN.1 인코딩 방식인 PER을 통해 최종 바이너리 형태의 Fast XML 문서가 생성된다.

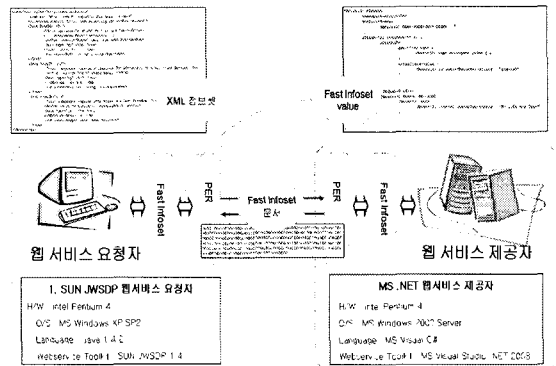


그림 9. Fast 웹 서비스 테스트 시나리오  
Fig. 9. Test scenario of Fast Web Services

웹 서비스 제공자 측의 시나리오는 웹 서비스 요청자가 보낸 Fast XML 파일을 다시 PER 방식으로 디코딩(decoding)하며, 디코딩으로 생성된 ASN.1 value를 Fast Infoset 시리얼라이징을 통해 원문의 XML 문서로 변환한다. 이때 웹 서비스 요청자가 보낸 XML 문서와 웹 서비스 제공자가 받은 XML 문서를 서로 비교하여 변환 전 문서

와 변환 후 문서의 이상 여부를 비교 판단해 평가한다.

그림 10은 Fast Infoset 처리를 위해 ASN.1 모듈에 따라 XML 정보셋을 재구성 한 모습을 보여준다. Fast Infoset 처리를 위해 어휘 테이블과 인덱싱 처리를 한 후 인덱싱 정보를 통해서 문서 정보를 단순화하는 중간 과정을 ASN.1 value로 표현한 것이다.

Fast Infoset 처리를 통해 얻어진 ASN.1 value는 PER 인코딩을 통해 최종적으로 바이너리화 되는데, 이 과정을 거쳐 인덱싱 정보만을 인코딩하여 파일의 크기를 줄인다.

```

<Title language="english">My Life and Times</Title>
choice(1) element {
  namespace-attributes { _ },           Element 정보 항목 처리
  qualified-name {
    choice(2) name-surrogate-index { 5 }
  },
  attributes sequence-of(1) {
    attribute {
      qualified-name {
        choice(2) name-surrogate-index { 6 }
      },
      normalized-value {
        choice(1) literal-character-string { 'english' }
      }
    }
  },
  children sequence-of(1) {
    choice(4) character-chunk {
      character-code {
        choice(1) literal-character-string { 'My Life and Times' }
      }
    }
  }
}
    
```

그림 10. 정보 항목 처리 결과  
Fig. 10. Processing result of Information Item

그림 11은 Book Element 정보를 Fast Infoset 처리한 결과를 XML Infoset과 비교하였다.

```

(01)00000001 : BookStore.child[1] 선택 = 1 (Book)
(B2)10110010 : Book의 길이는 178
(00)00000000 : Book.attribute-namespaces 개수 = 0
(02)00000010 : Book.qualified-name 선택 = 2 (index)
(01)00000001 : index의 길이는 1
(03)00000011 : index = 3
(01)00000001 : Book.attributes 개수 = 1
(08)00010000 : Book.attribute[1]의 길이는 8
(02)00000010 : Book.attribute[1].name 선택 = 2 (index)
(01)00000001 : index의 길이는 1
(04)00000100 : index = 4
(01)00000001 : Book.attribute[1].value 선택 = 1 (string)
(03)00000011 : string의 길이는 3
(61)01100001 ~ (31)00110001 : string = a01
(05)00000101 : Book.children 개수 = 5
(A2)10100010 : Book.children의 길이는 162
    
```

그림 11. Book Element의 PER 처리 결과  
Fig. 11. PER processing result of Book Element

Fast 웹 서비스의 구축은 SOAP 메시지와 WSDL의 바이너리 인코딩을 필요로 한다.

그림 12는 SOAP 메시지에 대한 ASN.1 value를 PER 인코딩 처리한 결과 일부를 보인다.

```

032E010D 00020103 00010604
(03)00000011 : paramRefComplexTypeSimple.children
개수 = 3
(2E)00110111 : paramRefComplexTypeSimple.children
의 길이는 46
(01)00000001 : paramRefComplexTypeSimple.child[1]
선택 = 1 (varBoolean)
(0D)00001101 : varBoolean의 길이는 13
(00)00000000 : varBoolean.namespace-attributes 개수 = 0
(02)00000010 : varBoolean.qualified-name
선택 = 2 (index)
(01)00000001 : index의 길이는 1
(03)00000011 : index = 3
    
```

그림 12. SOAP 메시지에 대한 PER 처리  
Fig. 12. PER processing of SOAP message

### V. 성능 평가 및 고찰

본 논문에서는 시뮬레이션에 사용되는 XML 문서들을 두 그룹으로 구분하였다. 한 그룹은 중복 빈도가 높은 문서들로서 정보 목록의 일정한 패턴이 반복되어 나열되며, 다른 그룹은 중복 빈도가 낮은 문서들로 반복되는 정도가 적은 XML 문서들이다.

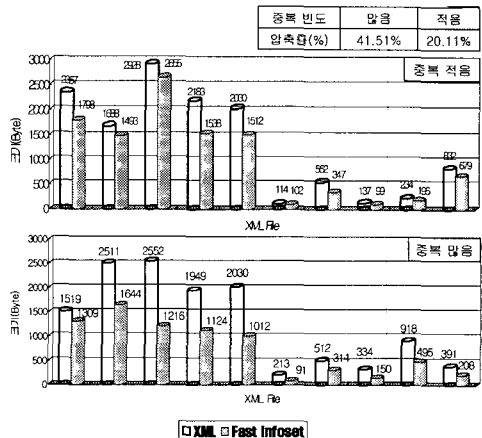


그림 13. Fast Infoset 처리 결과  
Fig. 13. processing result of Fast Infoset

그림 13은 시물레이션을 통해 변환된 XML 문서들의 크기를 비교한 것이다. 이 결과를 보면 중복빈도가 많은 문서들의 경우는 평균적으로 41.51%의 압축률을 보이며, 중복빈도가 적은 문서들의 경우는 20.11%의 압축률을 보이고 있음을 알 수 있다.

표 1은 그림 13의 결과 분석을 중복빈도와 문서크기에 따라 정리한 것이다.

표 1. 중복빈도와 문서크기에 따른 압축률  
Table. 1 Compressibility as repetition frequency and document size

문서크기 중복빈도	대	소	평균
적음	22.05%	19.51%	20.78%
많음	40.34%	43.88%	42.11%
평균	31.19%	31.69%	31.44%

결과적으로 문서의 구조에 의해 압축률의 차이는 다소 있었으나 테이블을 통한 중복제거와 같은 특징을 제외하더라도 시작 태그와 끝 태그가 사라지고 텍스트 데이터가 바이너리화 되기 때문에 기본적으로 Fast Infoset 방식의 인코딩이 XML 문서 압축에 효율적이라는 것을 알 수 있다.

## V. 결론

본 논문에서는 Fast Infoset의 방식과 ASN.1의 PER 인코딩 기법을 이용하여 XML 문서를 바이너리 형태인 Fast XML 문서로 변환 할 수 있는 Fast Infoset 알고리즘을 구현하였다. Fast Infoset 알고리즘은 XML 문서에 호환되는 바이너리 인코딩 알고리즘으로서, 이를 이용한 테스트 베드를 제공함으로써 다른 여러 어플리케이션에서도 이 모듈을 사용하여 테스트 할 수 있는 환경을 제공하였다.

구현된 Fast Infoset 인코딩 시스템은 모바일과 같은 적은 네트워크 대역폭에 사용이 용이할 뿐만 아니라 소형기기의 제한된 자원에서도 사용이 용이할 것으로 여겨지며, 구현된 시스템을 이용한 웹 서비스는 많은 응용분야에서의 핵심기술로 사용될 것이다.

또한, Fast Infoset 알고리즘의 웹 서비스 상호운용성

을 검증하기 위하여 바이너리 SOAP 메시지 전송을 테스트 할 수 있는 프레임워크를 구축하고, 전송된 메시지를 분석하고 검증하여 Fast 웹 서비스 개발자를 위한 가이드라인을 제시하였다.

향후 연구 과제로는 Fast Infoset 메시지를 웹 서비스에서 처리 할 수 있도록 ASN.1의 runtime library 모듈이 포함된 서버의 개발이 연구 되어야 할 것이다.

## 참고문헌

- [ 1 ] 변일수, 안창원, 박종근, 조희만, Desmons Frederic, 김성운, 바이너리 XML 기술 동향 2005.8.
- [ 2 ] XML Binary Characterization Working Group  
<http://www.w3.org/XML/Binary/>
- [ 3 ] David Hunter, Beginning XML, 2005.
- [ 4 ] XML 정보셋(Second Edition),  
<http://www.w3.org/TR/xml-Infoset/>
- [ 5 ] Fast Infoset@java.net, <http://www.idealliance.org/proceedings/xtech05/papers/04-01-01/>
- [ 6 ] ITU-T Rec. X.891 (2005) | ISO/IEC 24824-1, Fast Infoset
- [ 7 ] Abstract Syntax Notation One, <http://asn1.elibel.tm.fr/en/index.htm>
- [ 8 ] ITU-T Rec. X.680 (2002) | ISO/IEC 8824-1:2002, Specification of Basic Notation
- [ 9 ] Web Service Architecture, <http://www.w3c.org/ws-arch/>
- [ 10 ] ITU-T Rec. X.694 (2004) | ISO/IEC 8825-5:2004, Mapping W3C XML Schema Definitions into ASN.1
- [ 11 ] ITU-T Rec. X.690 (2002) | ISO/IEC 8825-1:2002, Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
- [ 12 ] ITU-T Rec. X.691 (2002) | ISO/IEC 8825-2:2002, Specification of Packed Encoding Rules (PER)
- [ 13 ] Objective Systems, Inc, <http://www.obj-sys.com>
- [ 14 ] JAX-RPC, JSR-101, <http://java.sun.com/xml/jaxrpc/>
- [ 15 ] JAXB, JSR-31, <http://java.sun.com/xml/jaxb/>

## 저자소개



**조 태 범(Tae-Beom Cho)**

1999년 배재대학교 정보통신공학과  
(공학사)

1993년 배재대학교 정보통신고공학과  
(공학석사)

2005년~현재 배재대학교 컴퓨터공학과 박사과정

1999년~현재 배재대학교 IT센터 전임강사

※관심분야 : XML, 웹서비스, 멀티미디어, 데이터베이스



**박 연 식(Yeoun-Sik Park)**

1971년 광운대학교 무선통신공학과  
(공학사)

1980년 건국대학교 행정대학원(행정  
학석사)

1999년 해양대학교 전자통신공학과(공학박사)

1970년~현재 경상대학교 정보통신공학과 교수, 해양  
산업연구소 연구원

※관심분야: 수중화상통신, 컴퓨터네트워크



**정 회 경(Hoe-Kyung Jung)**

1985년 광운대학교 컴퓨터공학과  
(공학사)

1987년 광운대학교 컴퓨터공학과  
(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)

1994년~현재 배재대학교 컴퓨터공학과 교수

※관심분야 : 멀티미디어 문서정보처리, XML, Web  
Services, SVG, Semantic Web, MPEG-21, Ubiquitous  
Computing, USN