

관계 데이터베이스 시스템 기반 XQuery 질의 처리기를 이용한 EDI 구축 시스템의 설계 및 구현*

김지선⁰ 홍의경
서울시립대학교 컴퓨터통계학과
{jskim98⁰, ekhong}@venus.uos.ac.kr

Design and Implementation of EDI Construction System using XQuery Processor based on Relational Database System

Jisun Kim⁰ Eui Kyeong Hong
Department of Computer Science and Statistics, University of Seoul

요 약

XML이 확장성과 문서 관계성의 우수성을 인정 받아 새로운 정보 공유 환경의 데이터 교환의 표준으로 각광 받으면서 기존의 EDI와 연동하여 XML을 사용할 수 있는 XML/EDI 시스템의 필요성이 대두되었다.

본 연구는 EDI 전자 문서를 관계 데이터베이스 시스템(RDBMS)을 통해 XML 문서 형식으로 저장하고 검색할 수 있게 하기 위해서 서울시립대학교에서 설계한 XML 문서 저장 기법을 사용하였으며, XPath, XQuery 질의를 SQL문 으로 변환시켜서 수행함으로써 EDI 전자 문서에 대한 검색 및 원하는 정보로만 구성된 EDI 문서 생성을 가능하게 하였다.

1. 서 론

급속한 인터넷 사용자 확산의 원동력이 된 웹(World Wide Web)은 이제 전자상거래를 위한 플랫폼으로 자리잡고 있다. 이에 따라 웹을 통하여 이기종 시스템에 구축되어 있는 문서들을 통합하고 교환하는 문제가 대두되었다. W3C에서는 SGML을 간편화시킨 XML을 제안하였다[1]. 기업 사이에 전자적 거래는 정보 통신을 이용하므로 확장성과 호환성을 만족하면서 경제적인 기술을 필요로 한다. XML은 확장성, 호환성이 좋으며 정보를 구조화하기에 용이한 특징을 갖고 있기에 이미 차세대 핵심 기술로 인정 받고 있다.

EDI(Electronic Data Interchange)는 이중의 거래 자원 시스템간의 자료 교환을 위한 표준으로 70년대부터 도입되어 대기업 중심으로 확산되었다. EDI를 통하여 기업간 메시지 교환, 비용 절감, 재고 관리 등을 효율적으로 처리할 수 있으나 시스템 구현을 위해서는 부가가치망(VAN)을 이용해야 하는 등 높은 고정 비용과 운영 비용이 든다[2]. 이에 반해 XML은 인터넷 기반으로 서비스를 제공할 수 있으므로 비용이 저렴하며 동적인 기업의 요구에 유연하게 대처할 수 있다.

XML이 새로운 데이터 교환 문서로 각광 받으면서 기존의 EDI와 연동하여 XML을 사용할 수 있는 XML/EDI 시스템을 필요로 하게 되었다. XML/EDI 시스템을 사용하는 기업이 특정 기업하고만 거래하지도 않을 뿐 아니라, 여러 다른 기업과 제휴를 맺게 되는 것이 일반적이다. 그러므로 데이터베이스에 저장된 정보에서 필요한 자료들만으로 EDI 문서를 만드는 일이 필요할 것이다.

따라서 EDI 문서를 작성하기에 필요한 데이터가 들어 있는 XML 문서를 관계 데이터베이스 시스템(RDBMS)에 효율적으로 저장하는 시스템을 개발하고, 사용자가 저장된 XML 문서에서 필요한 정보를 검색하여 그 결과를 EDI 문서로 볼 수 있도록 하기 위해 XQuery 질의어를 지원하는 연구의 필요성이 대두되었다. 본 논문에서는 서울시립대학교에서 설계한 XML 문서 저장 기법을 사용하였다.

본 논문의 구성은 다음과 같다. 2절에서는 XPath, XQuery 및 XML/EDI의 개념에 대하여 살펴보고, 3절에서는 논문에서 사용하는 XML 문서 저장 시스템의 구조에 대하여 설명을 한다. 4절에서는 XQuery 질의 처리기를 이용한 EDI 구축 시스템에 대하여 살펴본다. 마지막으로 5절에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

2.1. XML 관련 연구

2.1.1. XPath

XPath는 W3C에서 버전 1.0으로 권고하고 있으며[3] 문서에서 노드의 하위 집합을 선택하기 위해 단순한 구문을 제공하는 XML에 대해 정의된 질의어이다. XPath는 XML 문서의 일부분을 지시하기 위해서 사용한다. XPath는 URL 경로 표기법을 사용하여 XML 문서의 계층적인 구조를 논리적으로 탐색하는데 디렉토리나 유사한 경로뿐만 아니라 경로에 대한 조건을 지정함으로써 요소 집합을 검색할 수 있다.

본 연구에서는 XPath의 모든 기능을 만족하기보다는 XQuery에서 지원하는 생략된 표기식만을 사용하였다. 생략된 표기식과 비 생략된 표기식과의 비교는 [표 1]을 통해 확인할 수 있다.

표 1. XPath의 생략된 표기식

비 생략된 문법	생략된 문법
child::	(없음)
attribute::	@
/descendant-or-self::node()	//
parent::node()	..
(문서 루트 노드)	//

*본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

2.1.2 XQuery

XQuery는 XML 질의를 위한 새로운 W3C 표준으로서 W3C XML Query Working Group을 통해 정의되었다[4]. W3C XML Query Working Group이 가장 최근 발표한 XQuery 표준은 작업 초안 상태이기 때문에 XQuery 표준은 현재도 계속 발전하고 있다. XQuery는 XPath에 근거한 Path 표현과 For Let When Return(FLWR)로 표현된다. 이 논문에서는 중첩 질의를 배제하고 설계되었고, Path 표현은 2.1.1에서 설명한 XPath를 사용한다.

2.2. EDI 관련 연구

2.2.1. XML/EDI의 정의

EDI는 "기업이나 기관간의 전자적 문서교환 방식"[2]을 의미하며 기존의 문서교환방식을 대체하여 정보처리에 소요되는 노력을 절약하고 생산성과 효율성을 증대시키는 새로운 문서교환방식이다.

XML/EDI는 인터넷 기반의 EDI를 구현하기 위한 개방 표준의 한가지 접근 방법으로 XML/EDI 그룹에 의하여 제안되었다. 가격 및 제품 정보를 공급체인을 따라 교환하는 EDI는 XML을 적용할 수 있는 자연스러운 응용분야이다. 그 이유는 EDI가 이름이 부여된 구조적인 구성요소와 그 내용에 해당하는 데이터로 설계되어 있고, 이러한 구성요소는 XML의 DTD를 이용하여 정의할 수 있음은 물론 사전에 형식에 대한 동의 없이도 제품, 가격, 기타 속성 등을 기술하는데 필요한 구성요소를 수집할 수 있기 때문이다[5].

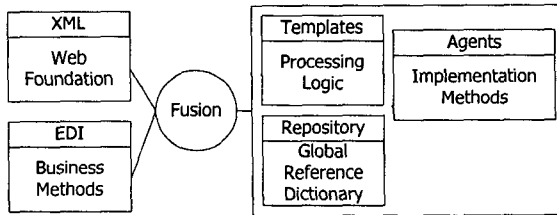


그림 1. XML/EDI의 요소 기술

[그림 1]은 XML/EDI에 필요한 5가지 구성 요소를 보여준다 [5]. 이 논문에서는 RDBMS에 저장되어 있는 정보에 XQuery 질의를 입력하고, 그 결과로 생성된 XML 문서로부터 EDI 전자 문서를 구축하는 XML/EDI 시스템을 설계 및 구현하였다.

3. XML 저장 시스템 구조

본 논문에서 native-XML 문서를 DTD에 상관없이 독립적으로 RDBMS에 저장할 수 있도록 하기 위해 [그림 2]와 같은 저장 스키마를 사용한다[6].

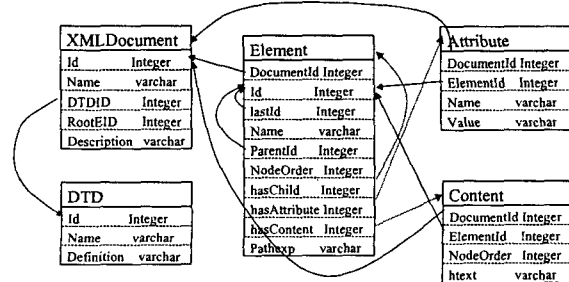


그림 2. XML 저장 스키마

XML 문서 검색은 XQuery 질의를 통해 이루어지며, XQuery 질의는 XPath를 기반으로 적절한 SQL문으로 변환된다.

본 시스템에서는 XML 문서를 저장하기 위해서 첫째로 XML 문서를 파싱하고, 이벤트를 이용하여 엘리먼트, 애트리뷰트, 내용 등의 정보를 각각의 테이블에 사상해서 저장한다

[그림 3]은 DFS Numbering을 이용하여 ID를 부여한 것하고 경로 정보를 추가적으로 표현한 그림이다.

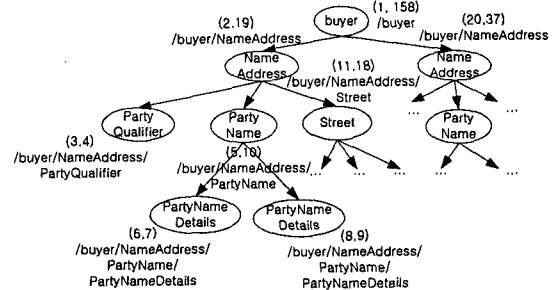


그림 3. DFS Numbering과 Pathexp 정보 표현 트리

[그림 3]에서 제시한 XML 문서에서 PartyNameDetails 정보를 찾기 위해 '/buyer/NameAddress/PartyName/PartyNameDetails' 라고 XPath 질의를 입력하면, 입력된 명령어는 토큰으로 분리된다. 각 토큰이 해당 문맥 노드를 구하는데 필요한지에 대한 정보를 확인하여 ContextNode와 Predicate정보로 분류한 뒤 SQL 질의가 생성된다.

```
SELECT e1.id startId, e1.lastId lastId, e1.parentId parentId,
       e1.xmlDocumentId xmlDocumentId
FROM   helement e1
WHERE  e1.xmlDocumentId = 1
AND    e1.name='PartyNameDetails'
AND    e1.pathexp = '/buyer/NameAddress/PartyName/PartyNameDetails'
```

본 과정을 통해 만들어진 SQL 질의문은 리턴 되어야 하는 엘리먼트의 범위 ID를 얻게 된다. XQuery 질의 처리기 에서 이 질의를 뷰로 만들어 이용한다.

4. XQuery 질의 처리기를 이용한 EDI 구축 시스템 설계 및 구현

EDI 문서를 구축하기 위한 XQuery 질의 처리기의 구조는 [그림 4]와 같다.

기업에서 XQuery 질의 처리기를 이용하여 EDI 문서를 생성한다면 거래하는 기업이 변경되거나 기업간의 거래 물품이 변경될 때마다 새로이 XML 문서를 만들지 않고, 데이터베이스에 저장되어 있는 정보로부터 필요한 내용을 생성해낼 수 있을 것이다.

EDI 문서에는 기업간의 거래에서 신청인과 공급자, 물품의 품목과 수량, 규격 등의 정보가 들어있다.

기업들의 정보가 저장되어 있는 데이터로부터 'AAA' 라는 이름을 가진 회사의 정보만을 추출하여 EDI 문서의 헤더에 '신청인'으로 삽입하기 위해서는 다음과 같은 XQuery 질의를 작성할 수 있다.

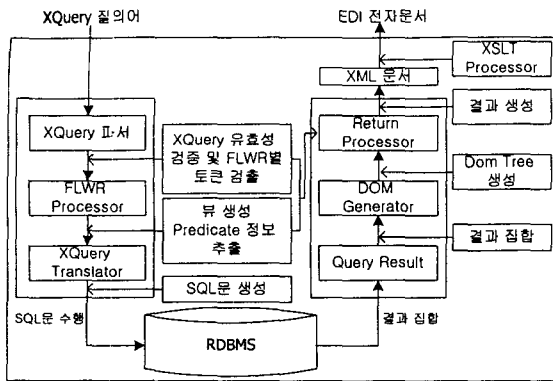


그림 4. XQuery 질의 처리기를 이용한 EDI 구축 시스템의 구조

```
FOR $b IN document("buyer.xml")/buyer/NameAddress
WHERE $b/PartyName/PartyNameDetails = "AAA"
RETURN <Purchase>
  <header>
    { $b }
  </header>
</Purchase>
```

XQueryParser 단계에서는 XQuery에 대한 유효성 검증과 FLWR 질 별로 토큰을 생성한다. FLWRProcessor 단계에서는 FOR 질의 변수 b에 대한 XPath 표현식에 대응하는 뷰 b_view를 생성한다. 그리고 WHERE 질에 대한 Predicate 정보인 \$b/PartyName/PartyNameDetails = "AAA"를 추출해 저장한다. XQueryTranslator 단계에서는 뷰와 Predicate 정보를 이용해서 다음과 같은 SQL문을 생성한다.

```
SELECT b_view.*
FROM b_view, element e1, content con1
WHERE b_view.xmldocumentid = e1.xmldocumentid
AND b_view.startId <= e1.id
AND b_view.lastId >= e1.lastid
AND e1.xmldocumentid = con1.xmldocumentid
AND e1.id = con1.elementid
AND e1.name = 'PartyNameDetails'
AND e1.pathexp = '/buyer/NameAddress/PartyName/PartyNameDetails'
AND con1.htext = 'AAA'
```

XQuery 질의의 결과로 생성된 XML 문서는 아래와 같다.

```
<Purchase>
<Header>
<NameAddress TagType="Segment" EDITagName="NAD">
  <PartyQualifier TagType="DataElement" ContainCharacter="all"
    MaxLength="3">MS</PartyQualifier>
  <PartyName TagType="DataElement">
    <PartyNameDetails TagType="CDATAElement"
      ContainCharacter="all" MaxLength="35">AAA
    </PartyNameDetails>
    <PartyNameDetails TagType="CDATAElement"
      ContainCharacter="all" MaxLength="35">Mr. Lee
    </PartyNameDetails>
  </PartyName>
  <Street TagType="DataElement">
    <StreetPOBox TagType="CDATAElement"
      ContainCharacter="all" MaxLength="35">XML Building
```

```
</StreetPOBox>
<StreetPOBox TagType="CDATAElement"
  ContainCharacter="all" MaxLength="35">Dongdaemungu
</StreetPOBox>
<StreetPOBox TagType="CDATAElement"
  ContainCharacter="all" MaxLength="35">Seoul
</StreetPOBox>
</Street>
</NameAddress>
</Header>
</Purchase>
```

생성된 XML 문서를 이용하여 구현된 EDI 전자문서는 다음과 같다.

```
NAD+MS+AAA:Mr. Lee:XML Building:Dongdaemungu:Seoul'
```

위의 EDI 문서는 일반적인 EDI 문서에서 신청인의 정보만이 들어있는 일부분만을 구현한 것이다. FLWR 질을 중첩하여 사용하면 신청인과 공급자, 그리고 판매 품목의 단가 등의 정보까지 포함한 완전한 EDI 문서를 구현할 수 있을 것이다.

5. 결론 및 향후 연구 방향

본 논문에서는 관계 데이터베이스 시스템에 저장되어 있는 정보로부터 XPath 또는 XQuery 질의를 이용한 결과를 토대로 EDI 문서를 구축하는 XML/EDI 시스템의 설계 및 구현 방법에 대하여 살펴보았다. 따라서 본 논문의 시스템은 기존에 설계되었던 DOM 인터페이스 또는 SAX 인터페이스를 이용하여 XML 문서를 EDI 문서로 변환하는 시스템[7]보다 더욱 향상된 기능을 지원한다.

XML/EDI에 관련된 표준 작업은 XML/EDI 그룹, W3C ECI 그룹, CEN/ISSS, CommerceNet X12/XML 작업 그룹, EEMA EDI 작업 그룹, Redix International Inc., Microsoft 등에서 활발히 진행 중이다[5]. XQuery 표준안과 XML과 EDI를 결합하기 위한 표준안이 정립된다면 더욱 정확하고 효율적인 검색 시스템을 설계하여 EDI 문서 정보의 무손실을 보장하는 XML 문서를 구현할 수 있을 것이다.

참고 문헌

- [1] W3C, Extensible Markup Language (XML) 1.0, <http://www.w3.org/XML>, 1998.
- [2] 한국전산원, "국내 EDI 활성화방안", 한국전산원, pp.1-8, 1997.
- [3] W3C, XML Path Language (XPath) 1.0, W3C Recommendation, <http://w3.org/TR/xpath>, 1999.
- [4] W3C, Xquery 1.0: An XML Query Language, W3C Working Draft, <http://w3.org/TR/xquery>, 2002.
- [5] 정준원, "차세대 EDI 표준화 동향 분석", 한국전산원, pp.80-81, 1999.
- [6] 고영기, 홍의경, "분할 저장 시스템에 적합한 XPath 질의 처리기 설계", 한국정보과학회 '02 가을학술 대회 논문집 29(2), pp.52-54, 2002.
- [7] 이재명, "전자상거래 시스템을 위한 XML을 이용한 EDI 구축에 대한 연구", 건국대학교, 2001.