

---

# XMDR을 이용한 정형화된 메시지 교환 기법 설계

황치곤\* · 정계동\* · 최영근\*

## Design of Formalized message exchanging method using XMDR

Chi gon Hwang\* · Kye dong Jung\* · Young-keun Choi\*

### 요 약

최근 들어 XML은 데이터 교환을 위한 표준으로 널리 사용되고 있으며, XML 문서의 크기가 커지는 경향이 나타나고 있다. 특히, 데이터 웨어하우스와 같이 대량의 데이터를 수집하여 분석하는 경우 데이터 이동은 트래픽의 증가로 인해 문제점이 발생할 수 있다.

따라서 이러한 문제를 해결하기 위해 XMDR 래퍼는 XML Schema의 트리구조를 분석하고, 분석된 트리 구조를 통해 XML Schema를 재생성하여 XMDR\_Query와 함께 각 스테이션에 전송한다. 결과로 반환되는 XML 문서는 XML Schema에 따라 XML 태그를 코드화 하여 정형화된 메시지를 전달한다. 정형화된 XML 문서는 네트워크의 트래픽을 감소시키고, XML 계층정보를 포함하고 있어 데이터 추출, 변환 및 정렬에 효율적이라는 장점을 가진다. 뿐만 아니라 정형화된 형식이므로 XSLT를 통한 변환과정에서도 효율적이다. 각 스테이션에 전송되는 XML Schema와 XMDR\_Query는 XMDR(eXtended Meta-Data Registry)을 통해 생성하고, 각 스테이션의 래퍼에서 결과 생성 및 XML 변환이 이루어지도록 하는 방법을 제안한다.

### ABSTRACT

Recently, XML has been widely used as a standard for a data exchange, and there has emerged the tendency that the size of XML document becomes larger. The data transfer can cause problems due to the increase in traffic, especially when a massive data such as Data Warehouse is being collected and analyzed.

Therefore, an XMDR wrapper can solve this problem since it analyzes the tree structures of XML Schema, regenerates XML Schema using the analyzed tree structures, and sends it to each station with an XMDR Query. XML documents which are returned as an outcome encode XML tags according to XML Schema, and send standardized messages. As the formalized XML documents decrease network traffic and comprise XML class information, they are efficient for extraction, conversion, and alignment of data. In addition, they are efficient for the conversion process through XSLT, too, as they have standardized forms. In this paper we propose a method in which XML Schema and XMDR\_Query sent to each station are generated through XMDR(eXtended Meta-Data Registry) and the generation of products and XML conversion occur in each station wrapper.

### 키워드

XMDR(eXtended Meta-Data Registry), XML, MDR, Ontology, Distributed Database

## I. 서론

컴퓨터 기술의 발전과 산업의 다변화로 인해 별도로 구축된 데이터베이스를 통합해야 하는 경우가 많이 발생하고 있다.

이에 기업의 각 조직과 주요 업무, 핵심 어플리케이션에서 발생하는 물리적인 데이터 소스들을 표준 규칙과 메타데이터를 이용하여 중복성이 제거된 데이터, 정확성이 보장된 데이터를 위해 데이터 통합이 필요하다[1]. 이런 데이터들을 효율적으로 통합하고 기존 시스템들은 그대로 유지하기 위해서는 이질적 환경, 의미적 이질성, 구조적 이질성 문제들이 나타난다.

이런 이유로 분산된 데이터베이스들을 통합할 필요성이 제기된다. 목적에 따라 데이터베이스를 업무에 적용시킬 수 있고, 이질적이고 자치성을 가진 데이터베이스들을 논리적으로 통합하여 클라이언트에게 투명한 정보 제공과 처리가 가능하도록 신뢰성, 안정성, 상호운용성의 문제를 해결해야 한다[2][3][4].

데이터 교환과 표준화를 위해 XML[5]은 가장 널리 사용되는 문서 형식이다. 그러나 XML의 성공은 XML이 의도되지 않은 곳까지 사용되게 되었고, 이에 따른 문제점이 거론되었다. 문제점 중 하나는 XML문서 사이즈가 커짐으로 네트워크의 부하가 가중된다는 점이다. 이는 소형 기기의 데이터 이동이나 데이터 웨어하우스의 대량의 데이터 이동에서 치명적인 약점이 될 수 있다. 이 문제를 해결하기 위해 XML의 양을 줄이기 위한 다양한 형식에 대한 연구가 진행되고 있다[6].

본 논문에서는 데이터 통합을 위해 XMDR을 이용하여 데이터의 구조적·의미적 이질성을 해결하고, XML 데이터의 전송 방법에 대해 기존 XML의 취지를 유지하면서 데이터 교환의 효율을 높이는 방안으로 XMDR을 기반으로 한 정규화된 메시지 교환 기법을 제안한다.

본 논문의 구성은 각 장별로 다음과 같이 구성된다. 2장은 관련연구로 본 논문의 기반이 되는 XMDR 구조와 생성방법, XML에 대한 다양한 연구에 대해 알아본다. 3장은 본 논문의 시스템에 대한 구성과 전송을 위한 XML 문서의 변환과 생성방법에 대해서 기술한다. 4장에서는 본 논문에 대한 평가를 수행한다. 5장에서는 결론과 향후과제에 대해 기술하도록 한다.

## II. 관련연구

### 2.1 XMDR

XMDR은 구조화된 메타데이터에 카테고리 분류, 온톨로지, 시소러스 등의 개념을 결합한 것이다[7]. XMDR 모델은 그림 1에서와 같이 Category, Knowledge Base, MDR로 구성된 XMDR Layer로 구성된다.

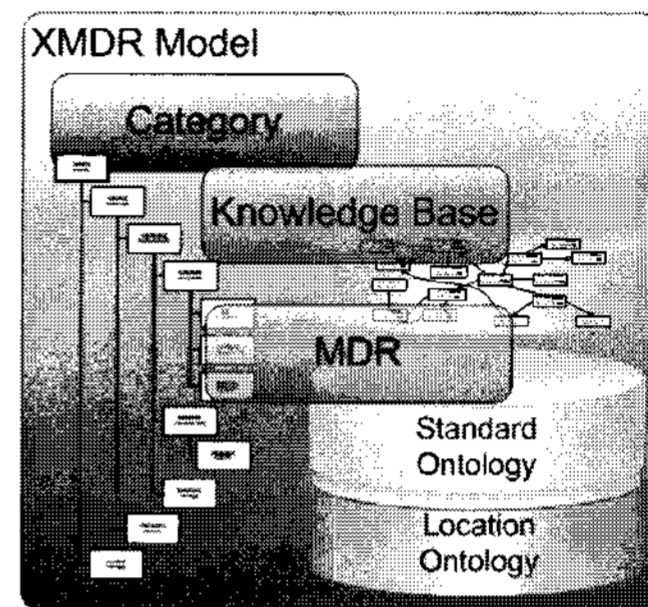


그림 1. XMDR 모델  
Fig 1. XMDR Model

- 카테고리(Category): 카테고리는 업무 구분에 따른 용어들에 대한 정의들로 상품과 서비스들에 대한 용어 및 의미가 정의되고, 공유되는 용어들의 집합을 트리로 표현한다.
- 지식 베이스(Knowledge Base): 지식 베이스는 온톨로지를 이용하여 구조화된 구체적 지식을 표현하고, XMDR 계층에서 정의된 클래스, 관계 등의 인스턴스들로 구성된다. 레거시 시스템에서는 상품 간의 관계 또는 서비스가 제공되는 지역 간의 관계성 등을 정의하고 이 계층에서 표현되는 개념 관계는 동등관계(Equal-Bridge), 상속관계(Is-Bridge), 부분관계(Part-Bridge), 대체관계(Substitute-Bridge)로 표현된다.
- MDR(Meta-Data Registry): MDR Layer는 표준 온톨로지와 로케이션 온톨로지 구성되는 계층으로 그림 1과 같이 XMDR의 구성한다.
- 표준 온톨로지: 표준항목, 표준항목과 레거시 시스템의 로컬항목 간의 매핑정보, 데이터타입간의 변환정보를 관리하는 역할을 한다.
- 로케이션 온톨로지: 레거시 시스템들의 위치 정보, 우선순위 정보, 접근 정보와 같은 정보를 관리하고 제공하는 역할을 한다.

이렇게 구성된 XMDR은 이질적 환경에서 데이터 표현, 저장 그리고 이동의 표준으로 사용되는 XML 문서로 통합하고, XML 스키마를 통해 XML 문서의 적합성을 검증한 표현이다. 본 논문에서는 이를 XMDR이라고 정의한다[8][9][10].

### 2.2 기존의 XML 변환 방법

XML의 사용범위가 확대되면서 발생한 문제점은 XML 문서의 크기의 증가로 인해 네트워크의 부하가 크다는 점이다. 이 문제를 해결하기 위해 XML 문서의 바이너리 형식에 대한 연구가 진행되기 시작하였다. 하지만 이것은 민감한 문제이다. XML 문서의 이진화는 XML의 본래 취지에 역행하는 것이라고 생각하는 사람도 적지 않다. XML의 개방성을 유지하면서 내재된 문제점을 해결하기 위해서는 바이너리 XML에 대한 표준이 필요하다[11].

웹 서비스는 이기종간의 상호운영성을 보장하기 위해 메시지 형식을 XML 형식으로 취하고 있다. 그러나 상호운영성에 대한 대가는 적지 않다. XML 형식의 메시지 교환은 XML의 내재된 단점으로 인해 전체 응용 프로그램의 성능을 저하시킬 정도로 큰 과부하를 일으키고 있다[12].

이 때문에 업계에서는 XML 문서의 바이너리 형식에 대해 요구하기 시작하였다. 이를 받아들여 W3C[13]에서는 바이너리 XML에 대한 정당성을 확인하기 위해 바이너리 XML에 대한 쓰임새를 분석하고 취합하는 워킹 그룹을 운영하고 있다[14]. 바이너리 XML의 이점은 분명하다. 데이터를 native 형식으로 저장함으로써 문서의 크기를 줄이고 XML 문서의 파싱 과정에 따른 과부하를 줄임으로 성능을 향상시킬 수 있다. 그러나 논란의 여지는 적지 않다. XML의 최대 장점은 상호운영성인데 이를 포기하고 바이너리 형식을 취한다는 것은 과거로의 회귀를 뜻하기 때문이다. 현재로서 이에 대한 가장 현실적인 접근 방법은 XML 문서의 바이너리 인코딩 표준을 마련하는 것이다.

## III. XMDR을 이용한 정형화된 메시지 교환 기법

### 3.1 XMDR 래퍼 구조

정형화된 메시지 교환 기법을 적용한 XMDR 래퍼 시스템은 그림 2와 같다. 네트워크에 연결된 각 스테이션

은 XMDR 래퍼를 가지고 있어 여러 스테이션에서 관리하는 데이터를 자신의 데이터처럼 사용할 수 있도록 하는 시스템이다.

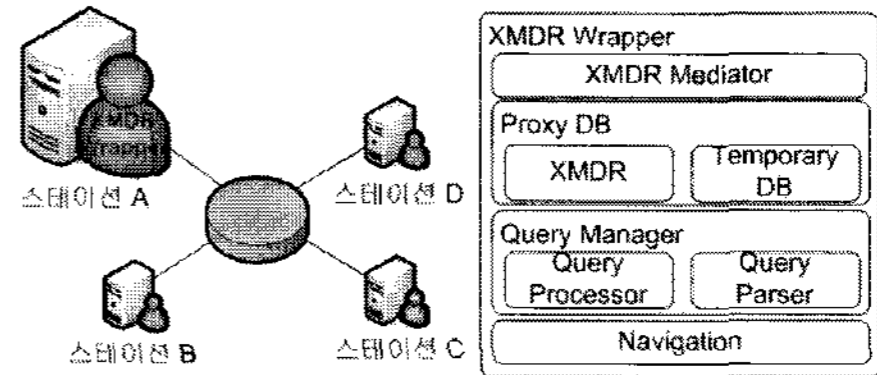


그림 2. 시스템 구조  
Fig 2. System structure

시스템의 래퍼는 그림2와 같이 구성되어 유기적으로 작업을 수행한다. 시스템의 각 구성 요소들에 대해서 알아본다.

XMDR 중개자(XMDR Mediator)는 XMDR을 통해 스키마 정보를 추출하는 역할을 수행하며 Query 관리자의 요구에 의하여 사용자가 인터페이스에서 선택하고 입력된 내용의 표준 항목을 제공하고 각 스테이션에서 변경된 스키마 또는 추가 되는 스테이션의 스키마를 이용하여 최신의 XMDR로 유지한다.

프록시 데이터베이스(Proxy DB)는 XMDR과 Temporary DB를 가지고 있다. XMDR은 표준항목과 각 스테이션에 의 로컬 필드명과의 매핑정보와 다른 스테이션의 위치 및 접근권한을 유지한다. Temporary DB는 다른 스테이션의 데이터베이스를 검색한 결과를 일시 저장하여 사용자에게 제공하는 역할을 한다. 자료 제공은 완료될 때까지 대기 하지 않고 특정 단위만큼 전송된 자료들을 먼저 사용자에게 확인시켜 주도록 하여 사용자의 대기 시간을 줄일 수 있도록 하고 갱신을 통한 후속 처리결과를 확인할 수 있도록 한다.

질의 관리자(Query Manager)는 XMDR 스키마 관리자에서 관리하는 표준 정보와 사용자의 입력정보를 통해 질의문을 생성하고 관리하는 역할을 수행하며, 질의 처리기(Query Processor)와 질의 파서(Query Parser)로 구분된다.

질의 처리기(Query Processor)는 파서에 의해 분해된 질의 정보와 XMDR 스키마 관리자에서 제공된 XMDR 정보를 통해 XMDR\_Query를 생성하는 역할을 수행한다. 질의 파서(Query Parser)는 XMDR 스키마 관리자로부터 제공되는 정보에 적합하도록 사용자의 입력정보

를 분석하여 분리하여 질의 처리기를 통하여 XMDR\_Query를 생성할 수 있도록 지원한다.

운영 에이전트(Navigation Agent)는 질의 처리기에 의해 생성된 글로벌 XMDR\_Query를 각 스테이션의 XMDR 래퍼에게 표준 질의를 전송하는 역할을 수행한다.

### 3.2 정형화된 XML의 변환

정형화된 XML은 XML 스키마를 분석하여 생성된 XML의 계층 구조에 대한 인덱스를 코드화하여 XML 태그로 사용함으로써 XML 문서를 정형화시킨 것이다. 데이터 이동이 정형화된 XML로 이루어짐으로써 네트워크의 트래픽 감소와 변환을 보다 효율적으로 수행할 수 있다. 사용자의 요구에 의해 XMDR의 정보를 제공받아 XMDR\_Query와 XML 스키마를 생성한다.

XML 스키마에 포함되는 인덱스 정보는 XML 스키마에서 element 요소의 계층 구조를 추출하고 계층 구조에 가중치를 부여한 것이다. 이 인덱스를 XML 스키마에 다시 부여하여 생성된 스키마를 정형화된 XML 스키마라 한다. 정형화된 XML은 정형화된 XML 스키마를 이용해 eid 식별자로 생성된다.

그림 3은 인터페이스에서 사용자의 요구를 받아 XMDR을 통해 다른 스테이션에 질의하기 위한 XMDR\_Query를 생성하고, 이와 동시에 XML 결과 형식인 XML 스키마를 생성하는 과정을 나타낸 것이다.

사용자의 요구 조건과 요구 항목을 XMDR에서 표준 항목과 테이블간의 관계성을 추출할 수 있다. 이를 통해 결과로 요구할 내용의 XML 스키마를 생성한다. 또한 XML 스키마는 계층구조를 가지므로, 계층 구조를 분석하여 계층 구조 Map을 생성하여 각 계층에 대한 고유한 식별자를 생성할 수 있다. 이 고유한 식별자를 eid라는 속성으로 XML 스키마에 포함시켜 다른 스테이션에 XMDR\_Query와 XML 스키마를 전송하여 요구사항과 결과에 대한 형식을 제공한다.

eid는 결과에 대한 최상위 element를 'r'로 부여하고 그 자손 element는 'r1, r2, r3, ...'순서로 부여한다. r1의 자손 element는 'r11, r12, r13, ...'와 같은 방법으로 부여한다. 이렇게 함으로써 계층상에서 같은 위치의 element를 찾고자 한다면 eid의 자리수만으로 검색이 가능하게 된다. 변환된 정형화된 XML 스키마는 그림 4와 같이 표현된다.

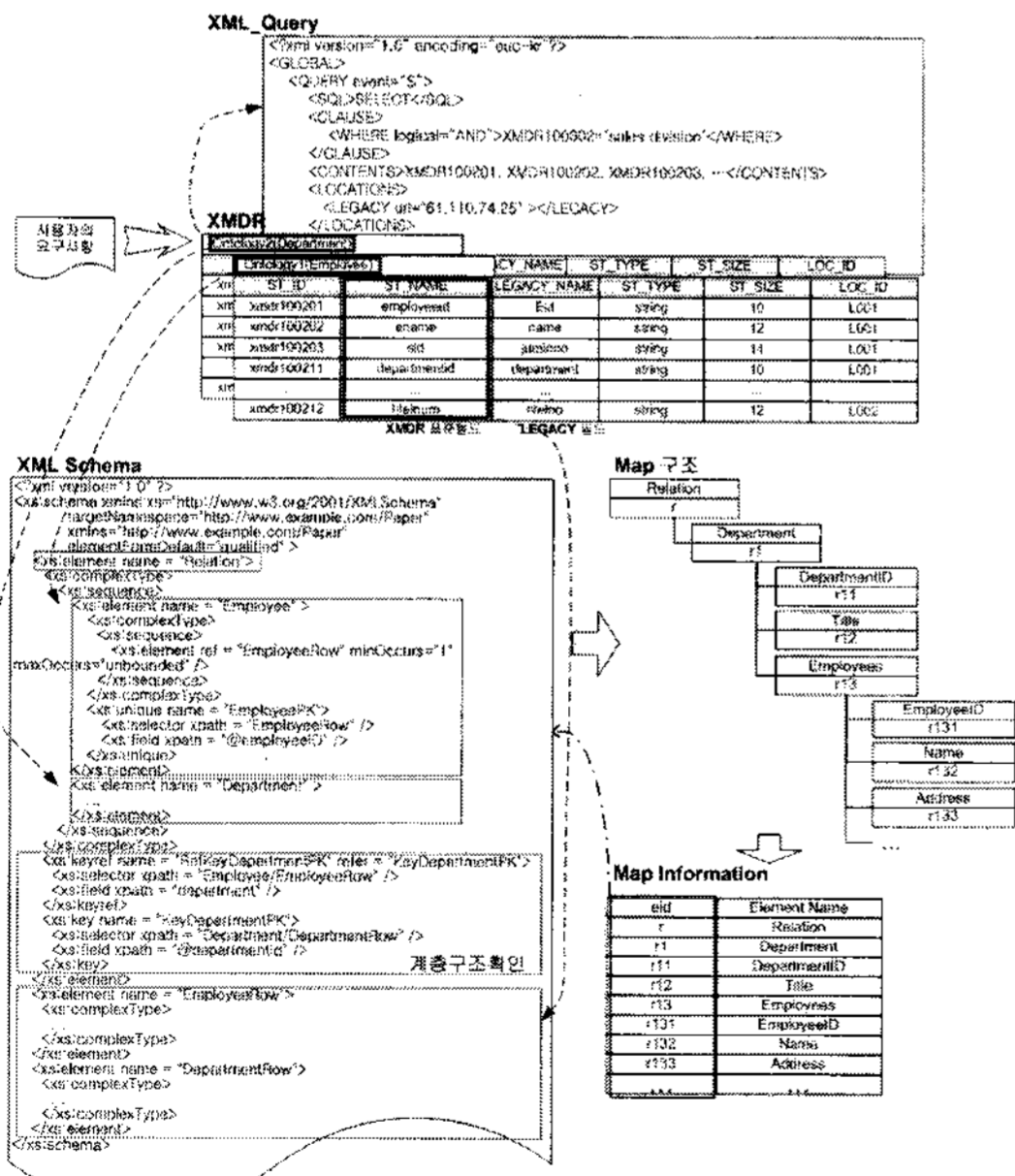


그림 3. XMDR 기반의 정형화된 XML 스키마 생성  
Fig 3. Creation of formalized XML Schema based on XMDR

```

<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.example.com/Paper"
xmlns="http://www.example.com/Paper"
elementFormDefault="qualified" >
<xs:element name="Relation" eid="r">
<xs:complexType>
<xs:sequence>
<xs:element name="Employees" eid="r13">
<xs:complexType>
<xs:sequence>
<xs:element ref="EmployeeRow" minOccurs="1"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:unique name="EmployeePK">
<xs:selector xpath="EmployeeRow" />
<xs:field xpath="@employeeID" />
</xs:unique>
</xs:element>
<xs:element name="Department" eid="r1">
...
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:keyref name="RefKeyDepartmentPK" refer="KeyDepartmentPK">
<xs:selector xpath="Employee/EmployeeRow" />
<xs:field xpath="department" />
</xs:keyref>
<xs:key name="KeyDepartmentPK">
<xs:selector xpath="Department/DepartmentRow" />
<xs:field xpath="@departmentid" />
</xs:key>
</xs:element>
<xs:element name="EmployeeRow">
<xs:complexType>
<xs:sequence>
<xs:element name="employeeid" eid="r131">

```

```

</xs:complexType>
</xs:element>
<xs:element name="DepartmentRow">
  <xs:complexType>
    ...
  </xs:complexType>
</xs:element>
</xs:schema>

```

그림 4. 정형화된 XML 스키마  
Fig 4. Formalized XML Schema

이렇게 XML 스키마를 정형화된 XML 스키마로 변환하는 과정을 알고리즘으로 표현하면 그림 5와 같이 표현할 수 있으며, 변환과정은 트리의 깊이 우선 검색 방법과 유사하다.

```

elementProc()
{
  ...
  readXMLSchema()//XML Schema 읽어온다.
  do{
    //스키마파일을 읽어 element 요소를 찾는다.
    seekElement()
    //element의 속성이 name이면
    if(is element and level==0)
    //루트 레벨 element이면 최상위 식별자(eid)를 부여한다.
    {
      eid='r'
      save element(eid, element_name)
    }
    else //루트 element가 아닐 경우
    {
      i=i+1
      eid=string_concatenate('r', i)
      //element 식별을 위한 식별자를 부여 받는다.
      save element(eid, element_name)
      //해당 element에 식별자(eid)를 부여한다.
      if(exist son_element)// 자식 element가 존재하면
      {
        save seekpoint //검색위치 저장
        seek_son_element(eid, element_name)
      }
    }
  }while(exist element)
  ...
}

void seek_son_element(char peid, char element_name)
{
  ...
  do{
    if(found element)
    {
      //자신인덱스:형제 element를 기준으로 순차적으로 부여 받는다.
      i=i+1
      eid=string_concatenate(peid, i)
      save element(eid, element_name)
      //해당 element에 식별자(eid)를 부여한다.
      if(exist son_element)// 자식 element가 존재하면
      {
        save seekpoint//검색위치 저장
        seek_son_element(eid, element_name)
      }
    }
  }while(exist sibling element)
  ...
}

```

그림 5. XML 스키마 변환 알고리즘  
Fig 5. XML schema converting algorithm

알고리즘은 XML 스키마를 읽어 XML 스키마의 element를 추출하여 그 속성이 name인지 ref인지를 확인한다. element의 속성이 name이면 map에 element 항목을 추가시키고, 속성이 ref이면 해당 참조영역이 지정한 element를 찾아 그 element의 하위 element를 map의 항목으로 추가한다. element 항목을 찾아 항목의 속성이 name이면 이전 element와 형제의 관계에 있으면 해당 eid를 증가시키고, 형제 관계가 아니라 계층상 자식 element이면 eid의 자리수를 늘여 이전 element와의 자식관계임을 표현하는 과정을 반복하여 XML 스키마의 계층정보를 eid로 추출하여 이를 다시 XML 스키마의 eid라는 속성으로 추가함으로 스키마의 변환과정을 가진다.

### 3.3 스테이션 간의 데이터 이동

생성된 정형화된 XML을 통한 데이터의 이동에 대한 경로와 전송되는 데이터를 본다. A 스테이션에서 사용자가 인터페이스에서 선택한 내용에 의해 XMDR의 정보를 추출하여 XMDR\_Query를 생성하고, 반환될 결과에 대한 XML 스키마를 생성하여 B 스테이션에게 네트워크를 통해 전송한다. B 스테이션에서는 XMDR\_Query를 분석하여 B 스테이션에 있는 데이터베이스를 검색하여 결과를 추출한다. 추출된 결과는 A 스테이션에서 전송된 XML 스키마에 정형화된 유효한 XML 문서로 생성된다. 생성된 정형화된 XML은 A 스테이션에 네트워크를 통해 전송한다. A 스테이션에서는 정형화된 XML을 temporary DB에 삽입하여 사용자의 요구에 따른 XSLT 변환을 거쳐 제공하거나 데이터 웨어하우스와 같이 분석을 위한 자료로 제공되도록 한다. 그리고 정형화된 XML은 XMDR에 의해 표준에 맞게 생성된 계층 정보에 의해 생성된 것이므로 더 이상의 변환과정은 없다. 이 과정은 그림 6와 같이 요약할 수 있다.

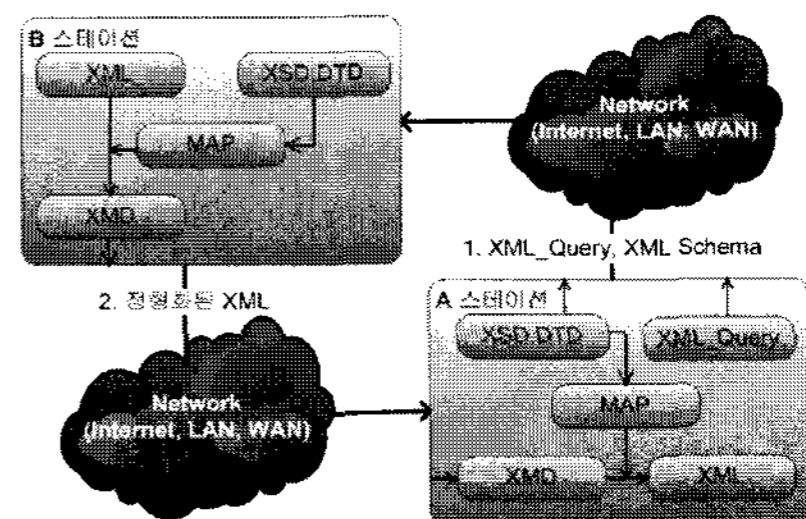


그림 6. 스테이션 사이의 데이터 이동  
Fig 6. Data transfer between a station

### 3.4 시스템 흐름

본 시스템의 작업 흐름은 그림 7과 같다. 사용자의 인터페이스 조작으로 질의 내용과 사용자 정보가 XMDR 래퍼로 전달되고 XMDR의 표준 정보와 지역 정보에 의해 생성된 XML 스키마와 XMDR\_Query를 요구된 스테이션에게 전송한다. 검색 요구를 받은 스테이션들은 XML 스키마에 유효한 XML 결과를 정형화된 XML로 검색 요구를 한 스테이션의 XMDR 래퍼로 전송하여 사용자에게 결과를 보여 준다. 각 단계의 작업은 다음과 같다.

- Query Item & UID : 사용자가 요구한 질의 내용 및 사용자 정보
- XMDR Search for Query Generation : XMDR을 이용하여 질의를 생성하기 위하여 표준 항목 정보 검색
- Query Information : XMDR의 표준 항목 정보 반환
- Location Information : XMDR의 각 스테이션 접근권한 및 정보
- Standard Query : 생성된 표준 질의를 Navigation Agent에 전달
- XMDR\_Query & XML Schema : 생성된 XMDR\_Query와 XML 스키마를 각 스테이션에게 전송
- 정형화된 XML : XML 스키마에 의해 변환된 정형화된 XML 문서 반환
- Result Query : 수집된 결과를 사용자 질의로 검색
- result : 요구 조건에 의한 검색결과

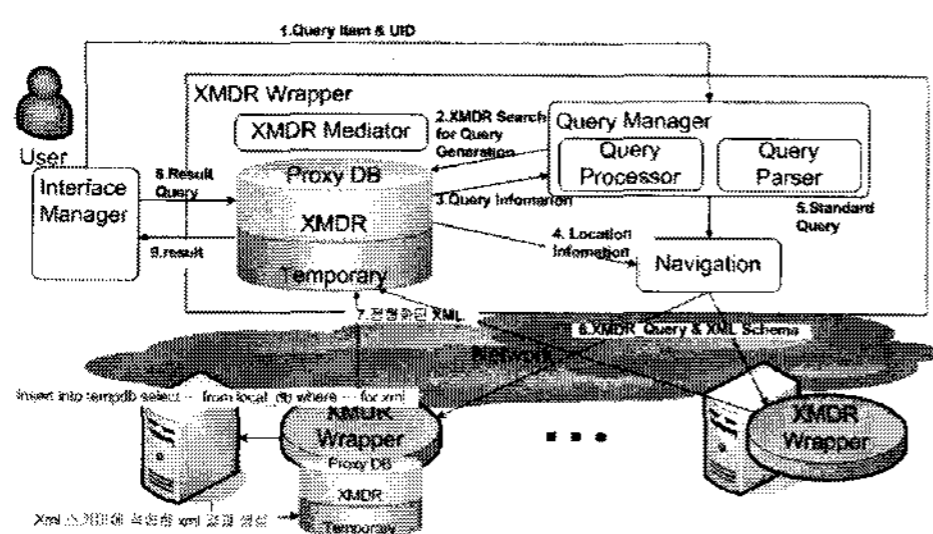


그림 7. 시스템 작업 흐름도  
Fig 7. Diagram of System work flow

### IV. 제안모델 적용 및 성능평가

본 논문에서 분산 환경에서 XMDR을 기반으로 XML 데이터의 전송 방법에 대해 기존 XML의 취지를 유지하

면서 정형화된 데이터 교환 기법을 제안하여 데이터 이동 효율을 높인다. 제안된 정형화된 메시지 교환기법을 위한 XML은 사원관리를 위한 employee테이블과 department테이블을 검색한 결과를 예제로 사용한다. 결과를 제안된 방법에 따라 변환시킨 전후의 데이터 변량에 대하여 정량적 테스트를 수행한다.

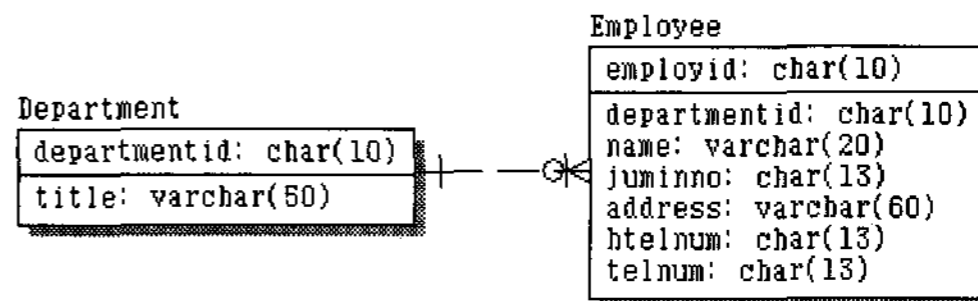


그림 8. 예제로 사용되는 테이블 구조  
Fig 8. Used as a sample table structure

테스트에 사용되는 테이블 구조는 그림 8과 같이 두 개의 테이블로 구성되고 이 두 테이블의 질의를 수행하고, 수행한 결과인 결과 XML을 비교한다.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Relation>
- <department>
  <title>기획부</title>
  <departmentid>1002</departmentid>
- <employee>
  <employid>10005</employid>
  <name>박남수</name>
  <juminno>7509081880033</juminno>
  <address>서울특별시 동대문구 신설동</address>
  <telnum>02-337-9870</telnum>
  <htelnum>019-765-7878</htelnum>
</employee>
- <employee>
  <employid>10007</employid>
  <name>김정주</name>
  <juminno>8812211234567</juminno>
  <address>서울특별시 노원구 월계동 123번지</address>
  <telnum>02-789-0987</telnum>
  <htelnum>010-8733-5670</htelnum>
</employee>
- <employee>
```

그림 9. 일반적인 XML  
Fig 9. General XML

```
<?xml version="1.0" encoding="utf-8" ?>
- <1>
  <11>1002</11>
  <12>기획부</12>
- <13>
  <131>10005</131>
  <132>박남수</132>
  <133>7509081880033</133>
  <134>서울특별시 동대문구 신설동</134>
  <135>02-337-9870</135>
  <136>019-765-7878</136>
</13>
- <13>
  <131>10007</131>
  <132>김정주</132>
  <133>8812211234567</133>
  <134>서울특별시 노원구 월계동 123번지</134>
  <135>02-789-0987</135>
  <136>010-8733-5670</136>
</13>
- <13>
```

그림 10. 정형화된 XML  
Fig 10. Formalized XML

질의 결과로 생성되는 XML은 그림 9과 같이 테이블과 필드로 구성되는 element와 실제 데이터인 값으로 구성되는 일반적인 XML이다. 그림 8의 결과는 부서별 사원의 상세내역을 확인 하는 것이다. 이러한 XML 문서를 정형화된 XML 스키마에 따라 변환시켜 XML 계층정보를 인덱스화한 eid를 이용하여 생성된 정형화된 XML로 XML 문서의 형식을 그대로 유지한다. 이는 그림 10와 같다. 정형화된 XML 문서로 변환했을 때 실제 용량이 얼마나 감소하는지 확인하기 위해서 우선 그림 8의 일반적인 XML과 정형화된 XML을 같은 형식의 데이터를 이용하여 테스트 수행했다. 수행 결과는 같은 데이터를 같은 형식으로 변환하였기 때문에 변환에 따른 감소율이 일정하게 나타났다. 표 1은 테스트한 결과의 수치이며, 이를 그래프로 표현하면 그림 11과 같이 일정한 추세로 정비례 관계를 나타내었다.

표 1. 같은 형식 데이터의 수행 결과  
Table 1. Execution results for the same type data

	1건	2건	3건	4건	5건	6건	7건	8건	9건	10건
변환전	373	605	837	1069	1301	1533	1765	1997	2229	2461
변환후	302	490	678	866	1053	1241	1429	1617	1805	1993

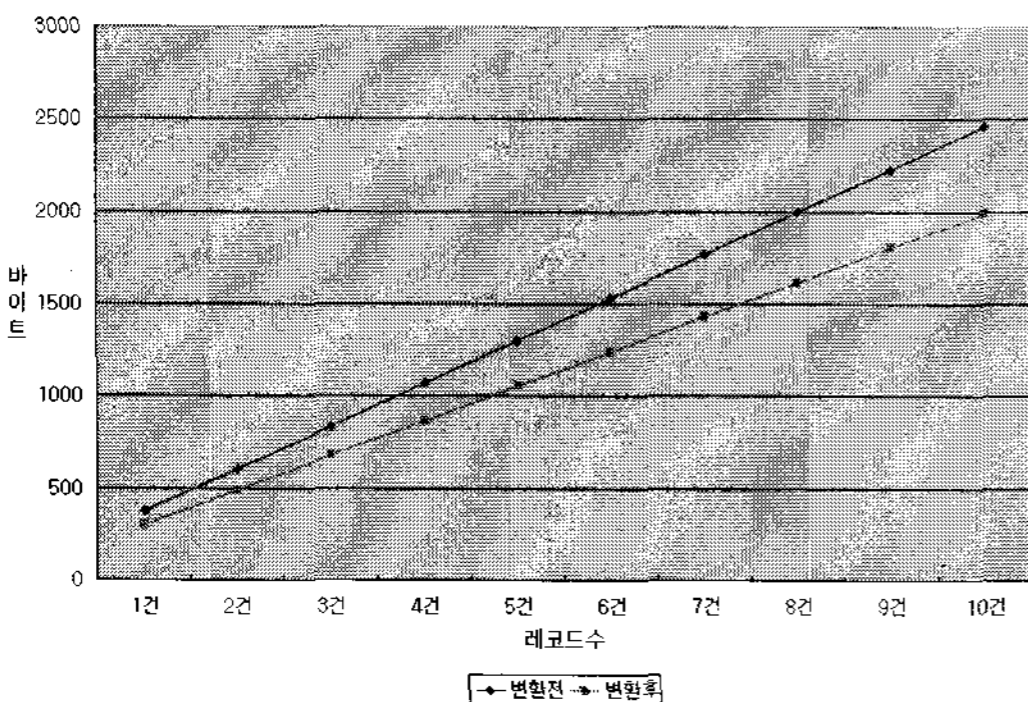


그림 11. 같은 형식 데이터에 대한 수행 결과  
Fig 11. Execution results for the same type data

그러나 적은 양이지만 그림 11의 그래프에서 볼 수 있듯이 변환전과 후의 데이터의 양은 레코드의 건수가 많아질수록 격차가 벌어지는 것을 볼 수 있다.

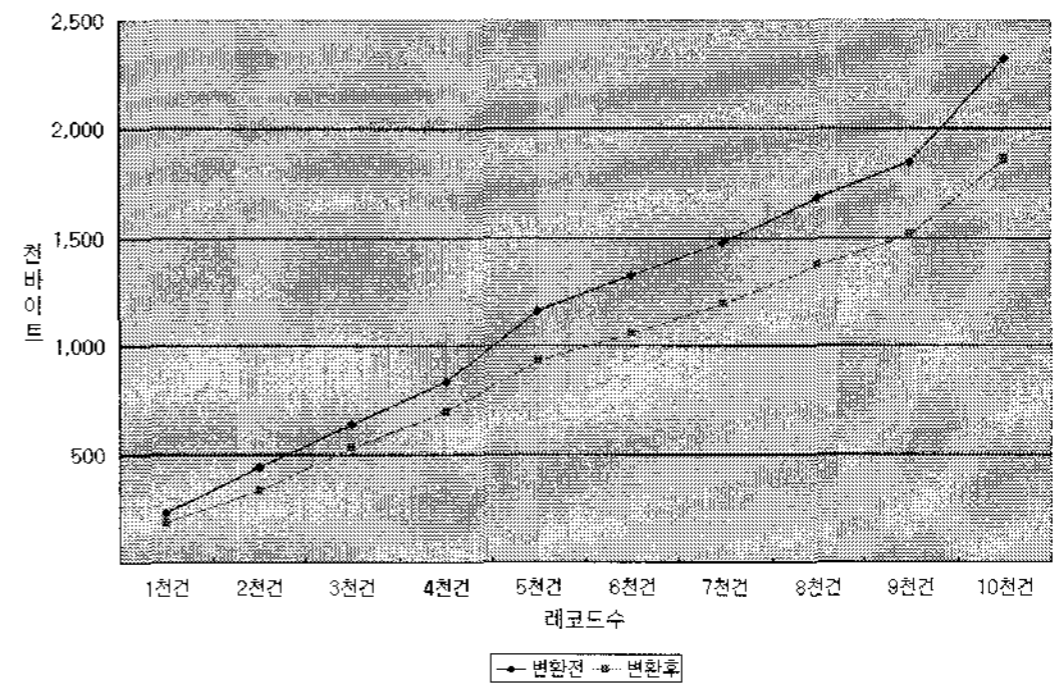


그림 12. 실제 데이터를 이용한 수행 결과  
Fig 12. Execution results for the real data

실제와 같은 조건하에서 테스트를 위하여 실제 데이터를 기준으로 천 건의 레코드에서 만 건의 레코드까지 수행결과를 확인하기 위해서 테스트한 결과는 그림 12과 같은 결과로 나타났다. 수행 결과는 완전히 소량의 레코드로 수행한 결과와 동일하게 레코드 건수에 따라 정비례 한다는 것을 확인할 수 있다. 그래프 상의 굴곡은 각 레코드의 필드 크기의 가변성과 필드의 특성 차이로 인해 발생한 것으로 전체적인 추세는 일정한 것으로 볼 수 있다.

테스트 수행 결과에 의하면 레코드 건수가 많아지거나 필드의 개수가 많아질 경우 데이터 변환에 따른 효율은 확대될 것이며, 네트워크상에서 데이터 트래픽의 감소 효과를 볼 수 있다.

## V. 결론

XML은 데이터의 이동에 대한 효율성 뛰어나 그 사용 범위가 원래 XML의 취지 보다 확대되고 있다. 사용 범위의 확대됨에 따라 적게는 소형통신기기나 모바일 통신기기에서, 크게는 데이터 웨어하우스의 데이터 수집에 까지 다양한 영역에서 사용되고 있다. 이러한 XML 데이터 전송은 많은 데이터의 양으로 인해 성능저하를 가져올 수 있다.

이에 대해 XMDR을 기반으로 생성한 XML 스키마에 XML의 계층 정보를 포함시켜 정형화된 XML을 이용함으로써 전송되는 데이터 양을 감소시켜 네트워크상의 트래픽 효율을 향상시키고자 했다. DBMS의 논리적 이질성과 물리적 이질성의 문제를 해결하기 위해 XMDR을

이용했다.

결과인 정형화된 XML은 XML의 계층정보를 포함하고 있어 데이터 검색, 추출, 변환 및 정렬에 효율적으로 적용할 수 있다. 계층 정보를 인덱스로 이용함에 따라 동일 인덱스에 대한 수평 검색과 계층 정보에 의한 수직 검색이 가능해짐으로 효율적이다. 뿐만 아니라 XSLT를 통한 변환과정에서도 XML 스키마에 따른 eid를 element로 사용하였기 때문에 XSLT 변환에도 추가 작업 없이 수행 가능하다는 장점이 있다.

이후 연구는 정형화된 XML 태그를 인덱스로 적용하기 위한 구체적인 방법론을 제시하고 실제 적용 사례를 통한 실제적인 효율을 검증하도록 하겠다.

### 참고문헌

- [1] 나정옥, "oracle data hub 실시간 기업을 위한 데이터 통합", Oracle Korea Magazine 2005 Summer, 2005.
- [2] A.S. Tanenbaum and M. Van Steen, "Distributed Systems:Principles and Paradigms.", Prentice Hall, 2002
- [3] P. Verissimo and L. Rodrigues, "Distributed Systems for System Architects.", Kluwer Academic Publishers, 2001]
- [4] 이승용, 박재복, 김명희, 주수종, "분산환경에서 혼용류 관리기법을 채택한 이질적인 멀티데이터베이스 상호운용 모델 설계", 정보처리학회논문지 D 제12-D 권 제4호, 2005.8
- [5] Extensible Markup Language, <http://www.w3.org/XML>.
- [6] XML Binary Characterization Working Group, <http://www.w3.org/XML/Binary/>.
- [7] xmdr project, <http://xmdr.org>.
- [8] 황치곤, 정계동, 최영근, "지식 공유 기반의 XMDR을 이용한 적응형 검색 시스템 설계", 한국통신학회논문지 제31권 제8B호 p.716-729, 2006.
- [9] 정계동, 황치곤, 최영근, "분산 환경에서 XMDR 기반의 멀티데이터 베이스 상호운영 모델 설계", 한국해양정보통신학회논문지 Vol.11 No.9 pp.1771-1780, 2007
- [10] 정계동, 황치곤, 최영근, "분산 환경에서 XMDR을 이용한 예약 정보 시스템", 한국해양정보통신학회 논문지 Vol.11 No.10 pp.1957-1967, 2007

[11] 변일수, 안창원, 박종근, 조희남, Desmons Frederic, 김성운, 바이너리 xml 기술 동향, ITFIND 주간기술 동향 1208호, 2005.08.10.

[12] Fast Web Service, <http://java.sun.com/developer/technicalArticles/WebServices/fastWS/>.

[13] World Wide Web Consortium, <http://www.w3c.org/>.

[14] XML Binary Characterization Working Group, <http://www.w3.org/XML/Binary/>.

### 저자소개

황 치 곤(Chi-gon Hwang)



1995년 창원대학교 경영학과(학사)  
2004년 광운대학교 정보통신학과  
(공학석사)

2008년 ~ 현재 광운대학교 컴퓨터 과학 박사과정

2006년 ~ 현재 전자넷 연구원

※관심분야: 웹서비스, XMDR, 그리드컴퓨팅, 이동에이전트, 상호운용

정 계 동(Kye-dong Jung)



1985년 광운대학교 전자계산학  
(이학사)

1992년 광운대학교 산업정보학  
(이학석사)

2000년 광운대학교 컴퓨터과학(이학박사)

1993년 ~ 2004년 광운대학교 정보과학원 교수

2005년 ~ 현재 광운대학교 교양학부 교수

※관심분야: XML 분산시스템, 분산 컴퓨팅기술, 이동에이전트

최 영 근(Young-keun Choi)



1980년 서울대학교 수학교육과  
(이학사)

1982년 서울대학교 계산통계학과  
(이학석사)

1989년 서울대학교 계산통계학과(이학박사)

1982년 ~ 현재 광운대학교 컴퓨터과학과 교수

1992년 ~ 2000 광운대학교 전산정보원 원장

2002년 ~ 2005 광운대학교 교무연구처장

※관심분야: 객체지향 설계, 분산시스템, 이동에이전트, 상호운용