

# 메타데이터 이질성 해결을 위한 MDR 기반의 메시지 변환 시스템

(A Message Conversion System based on MDR for  
Resolving Metadata Heterogeneity)

김진관<sup>†</sup> 김중일<sup>\*\*</sup> 정동원<sup>\*\*\*</sup> 백두권<sup>\*\*\*\*</sup>  
(Jin-Kwan Kim) (John I Kim) (Dongwon Jeong) (Doo-Kwon Baik)

**요약** 메타데이터는 데이터의 의미, 표현 등을 명확히 기술함으로써 공유 및 교환을 향상시키기 위한 데이터에 대한 데이터이다. 그러나 다양한 방식으로 생성된 메타데이터는 메타데이터간의 불일치라는 또 다른 문제를 야기하였다. 최근 메타데이터 불일치 문제를 해결하기 위하여 메타데이터 게이트웨이 방식에 대한 연구가 활발히 진행되고 있다. 그러나 메타데이터 게이트웨이 방식으로 구현된 기존의 시스템들은 메타데이터 스키마에 종속되어 메타데이터의 변화에 따른 시스템의 유지 보수에 많은 시간과 비용이 소요된다.

이 논문에서는 기존의 메타데이터 게이트웨이 방식이 가지고 있는 단점을 개선하기 위하여, 이질적인 메타데이터의 사상 정보와 사상 규칙을 분리한 개념을 적용한 메시지 변환 시스템을 제안한다. 이 논문에서 제안하는 시스템은 ISO/IEC 11179를 적용하여 표준화된 데이터 요소를 동적으로 관리하며, 향후 생성될 데이터 요소에 대한 표준을 제공함으로써 추가적인 메타데이터 불일치 발생 문제를 근본적으로 해결할 수 있는 기능을 제공한다.

**키워드** : 메타데이터, 메시지 변환, 데이터 공유, 데이터 교환, 메타데이터 이질성, XML(eXtensible Markup Language), MDR(Metadata Registry)

**Abstract** Metadata is a general notion of data about data to improve data sharing and exchanging by definitely describing meaning and representation of data. However, metadata has been created in various ways and It caused another kind of heterogeneity problem named metadata heterogeneity problem. Recently, the research on metadata gateway approach that allows metadata heterogeneity is being more actively progressed. However, the existing commercialized systems that have been implemented with the metadata gateway approach are dependent on a metadata schema. In this paper, we propose a message conversion system which separates the mapping information from the mapping rules between heterogeneous metadata schemas. The proposed system dynamically manages standardized data elements by applying ISO/IEC 11179. Therefore, the proposed system provides the set of standard data elements to create consistently metadata of new databases and provides a fundamental resolution to the metadata heterogeneity problem

**Key words** : Metadata, Message conversion, Data sharing, Data exchange, Metadata heterogeneity, XML(eXtensible Markup Language), MDR(Metadata Registry)

## 1. 서론

데이터의 교환은 데이터의 수신자가 송신자의 의도와 동일하게 정보를 해석하는 것을 보장하는 환경에서 이루어진다[1]. 메타데이터는 데이터의 의미(Semantic), 구조(Structure), 그리고 표현(Representation) 정보를 포함하고 있으므로, 이를 이용하면 데이터의 실제 의미를 파악할 수 있다. 그러나 메타데이터 표현 방식의 일관성을 보장해주기 위한 표준화된 규정의 부재로 인해 메타데이터간의 불일치 문제가 발생하였다.

메타데이터간 불일치를 해결하기 위해서 최근 메타데

<sup>†</sup> 비 회 원 : 고려대학교 컴퓨터학과  
jkwan@netian.com

<sup>\*\*</sup> 비 회 원 : (주)라임미디어테크놀러지스 기술이사  
jikim@limemedia.co.kr

<sup>\*\*\*</sup> 장 회 원 : 고려대학교 컴퓨터학과 교수  
withimp@software.korea.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 고려대학교 컴퓨터학과 교수  
baikdk@chol.com

논문접수 : 2003년 9월 18일

심사완료 : 2004년 3월 16일

이타간 이질성을 허용하는 메타데이터 게이트웨이 방법에 대한 연구가 활발히 진행되고 있다. 그러나 기존의 메타데이터 게이트웨이 방식은 메타데이터 스키마에 종속되어 메타데이터 스키마의 변화가 발생할 경우 시스템을 재구축해야만 하는 단점을 지닌다.

이 논문에서는 기존 메타데이터 게이트웨이 방식의 문제점을 해결하기 위하여 사상 정보와 사상 규칙의 독립성을 보장하는 메시지 변환 시스템을 제안한다. 제안하는 시스템은 메타데이터 레지스트리(Metadata Registry: MDR), MSDL (Metadata Semantics Description Language), 메시지 변환기로 구성된다. MSDL은 변환에 필요한 사상정보를 메시지 변환기에 정의된 사상 규칙과 분리하여 관리하는데 이용된다. MDR은 ISO/IEC 11179를 적용하여 표준화된 데이터 요소를 동적으로 관리함으로써 데이터 실제 의미 전달을 위한 기준 데이터 요소를 제공한다.

논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 기술하고, 3장에서는 메타데이터간 불일치 해결을 위한 사상규칙을 정의한다. 4장에서는 메타데이터간의 차이점을 기술하기 위한 마크업 언어인 MSDL을 정의하고, 이를 이용한 XML 메시지 변환 시스템에 대해 상세히 언급한다. 5장에서는 시스템의 구현 및 평가 내용에 대하여 기술하고, 6장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

## 2. 관련 연구

### 2.1 메타데이터 불일치 분류

이 절에서는 메타데이터 관련 용어에 대한 혼란을 피하기 위해 이 논문에서 사용하는 주요 용어에 대하여 먼저 정의한다. 메타데이터는 데이터를 설명하기 위한 구조화된 데이터로써 의미한다. 데이터 요소는 데이터의 특징을 나타내는 일반적이고 추상적인 개념으로 하나의 데이터를 나타내기 위한 메타데이터 단위이다. 메타데이터 스키마는 서로 연관을 가진 특정 데이터 요소들의 집합을 의미한다.

일반적으로 메타데이터는 데이터에 대한 정보를 전달하기 위해 표 1과 같이 의미, 표현, 구조의 3가지 요소를 지닌다[1]. 데이터의 원활한 공유를 위해서는 이러한 요소의 일관성 있는 기술이 요구된다. 그러나 메타데이터 생성을 위한 표준화된 규정의 부재로 인해 메타데이터간의 의미, 표현, 구조 불일치가 발생하였다[2]. 이러한 문제를 해결하기 위해서는 메타데이터 불일치에 대한 상세한 분류가 선행되어야만 한다. 이에 따라 메타데이터 불일치의 분류에 대해 그동안 많은 연구가 진행되어왔다.

Gio Wiederhold는 정보의 지능적 통합을 위해 이질

표 1 메타데이터가 가지고 있는 정보의 구성요소

의미 정보 (Semantic Information)	메타데이터가 가지고 있는 데이터 요소의 의미 정보
구조 정보 (Schematic Information)	각 데이터 요소간의 연관성을 나타내는 구조 정보
표현 정보 (Representation Information)	값 영역, 데이터 형, 측정 단위 등의 표현 정보

데이터 결합시에 발생할 수 있는 다양한 레벨의 이질성을 상향식으로 분류하였다[3]. W.Kim과 J.Seo는 CDB (Component Database)간의 통합시에 일어날 수 있는 스키마와 데이터 이질성을 정의하고 이를 크게 스키마 충돌과 데이터 충돌의 두 가지로 비교하였다[4]. Christine과 Stefano는 데이터베이스 통합시의 발생하는 문제점과 이를 해결하기 위한 접근법이라는 측면에서 발생하는 충돌 문제에 대하여 분류하고 해결책을 제시하였다[5]. Richard는 이질적인 메타데이터를 표현하기 위한 방법인 테이블, 규칙, 온톨로지, 모델에 대하여 장점과 단점을 비교 연구하였다[6].

### 2.2 메타데이터 불일치 해결을 위한 선행연구

메타데이터 불일치를 해결하기 위한 선행연구는 크게 단일 메타데이터 표준을 제정과 메타데이터 게이트웨이를 이용한 접근방법의 두 가지로 분류할 수 있다[2]. 단일 표준 제정 방법의 경우, 현실적으로 모든 데이터들을 수용할 수 있는 단일 메타데이터 표준을 정의하는 작업은 불가능하다는 문제점을 지닌다. 이러한 문제점을 해결하기 위하여 메타데이터간 이질성을 허용하는 메타데이터 게이트웨이 방법이 제안되었으며 최근 이에 대한 연구가 활발히 진행되고 있다. 그러나 BizTalk[7]과 같은 기존의 메타데이터 게이트웨이는 데이터 요소간의 사상 정보가 시스템에 종속되어 있다. 이러한 경우 메타데이터 스키마의 변경시 시스템도 함께 변경해주어야 하며, 사상된 메타데이터의 변화가 빈번한 경우 시스템의 유지보수에 많은 비용과 노력이 소요되는 단점이 있다. 이러한 문제점을 해결하기 위해서는 메타데이터간의 사상과 시스템이 분리된 형태의 시스템 개발이 필요하다. 즉, 사상 규칙과 사상 정보간의 독립성이 보장되는 시스템이 요구된다.

## 3. 메타데이터간 차이점 보정을 통한 불일치 해결 방법

이 장에서는 메타데이터 레지스트리를 통한 데이터 요소의 의미 동등성 구축 방법에 대하여 기술한다. 또한 데이터 요소간에 발생하는 표현 및 구조 불일치 해결을

위해 정의한 사상 규칙에 대하여 기술한다.

**3.1 MDR을 통한 의미 공유와 불일치 분류**

ISO/IEC 11179에서 제안하는 메타데이터 레지스트리(Metadata Registry: MDR)는 표준화된 메타데이터를 유지·관리하며, 메타데이터의 명세와 의미의 공유를 목적으로 한다[8].

MDR에 의해 관리되는 데이터의 기본단위는 데이터 요소이다. 하나의 데이터 요소는 하나의 의미를 나타내기 위해서 사용되며, 각 데이터 요소는 유일한 식별자를 가지고 있다. 따라서 MDR은 데이터 요소를 통해 사용자에게 데이터에 대한 유일하고 정확한 의미를 제공함으로써, 데이터 요소간의 의미상 동등성을 확보할 수 있다.

의미상의 동등성을 확보한 데이터 요소간에 일어날 수 있는 불일치로는 표현 및 구조의 불일치가 있다[9]. 이 논문에서는 기존에 연구된 데이터 불일치에 대한 분류를 동등한 의미를 갖는 메타데이터 간에 일어날 수 있는 불일치 문제로 재분류하고 그 유형을 정리하였다. 표 2는 동등의미 메타데이터 간에 발생할 수 있는 구조 및 표현의 불일치 유형을 요약한 내용이다. 표 2에서 분류한 불일치 유형들은 단독으로 존재할 수도 있지만 일반적으로 여러 유형들이 혼합되어 나타난다[4].

**3.2 표현 불일치 해결을 위한 사상 규칙**

의미상 동등한 데이터 요소들간에 교환을 위해서는 해당 요소 간의 표현 불일치 해결이 요구된다. 표현 불일치는 데이터형 변환 여부에 따라, 크게 코드셋을 사용하는 경우와 그렇지 않은 경우로 분류할 수 있다.

(1) 코드셋의 변환

코드셋을 사용하는 경우는 변환대상 코드셋과 MDR의 기준 코드셋간의 코드를 연관시켜 함께 나열하고, 각각의 코드를 치환함으로써 종류가 다른 코드셋을 변환할 수 있다. 다음은 코드셋 A와 B간의 코드셋 변환을 위한 규칙을 나타낸 것이다.

$$\text{코드셋 } A = \{x_0, x_1, x_3, \dots, x_n\},$$

코드셋  $B = \{y_0, y_1, y_3, \dots, y_m\}$  일 때,

$A, B$ 의 모든 원소에 대해 코드  $x_n$ 과  $y_m$ 이 의미상 동일한 의미의 코드이면,  $x_n$ 을  $y_m$ 으로 치환한다

(2) 데이터형과 측정단위의 변환

코드셋을 사용하지 않는 경우는 데이터형과 측정단위의 변환이 필요하다. 데이터형은 표현하는데 사용한 데이터형과 실제 의미의 차이에 따라 표현형식과 의미형식으로 분류된다. 데이터형의 변환은 강제형변환을 통해 수행되며, 의미의 전달이 가능한 데이터형 간에만 변환이 이루어진다. 측정단위의 변환은 데이터가 측정단위를 가지고 있는 수치데이터인 경우에 데이터 형식을 변환하는 과정에서 함께 이루어진다. 그림 1은 문자형으로 표현된 10cm를 문자형으로 표현된 10mm로 변환하는 경우의 예를 도식화한 것이다.



그림 1 데이터형과 측정 단위의 변환 예

**3.3 구조 불일치 해결을 위한 사상 규칙**

구조를 불일치 해결을 위한 사상 규칙은 치환, 합성, 분해, 재배치의 네 가지로 분류되며 표 3에 정의한 기본 명제를 기반으로 기술한다. 기본 명제는 KIF (Knowledge Interchange Format)[10]를 이용하여 표현하였다.

표 3 Function 정의를 위한 기본 명제

```

same(x, y) ← x = y
getElement(S, a) ← extract a from S
setElement(S(x), a) ← link a to S(x)
z = +(x, y) ← z means x and y
    
```

표 2 동등의미를 갖는 메타데이터간의 불일치 분류

분류	불일치 유형	설명	예
구조	합성 (Composition)	하나의 데이터 요소가 다수의 요소의 합집합에 대응	(FirstName U LastName) → Name
	분해 (Decomposition)	하나의 데이터 요소가 다수의 데이터 요소로 분리되어 표현	Name → (FirstName U LastName)
	재배치 (Rearrangement)	데이터 요소의 순서(Order) 정보가 필요하며, 그 배치가 다른 경우	FirstName, LastName → LastName, FirstName
표현	코드셋 (Code Set)	적용 코드셋의 불일치	ISO 3166-2 ↔ ISO 3166-3
	측정단위 (Measurement Unit)	측정 단위의 불일치	Mile ↔ Kilometer
	데이터형 (Data Type)	데이터형의 불일치	Integer ↔ Float

표 3의 기본 명제는 다음과 같은 의미를 지닌다. 데이터 요소  $x$ ,  $y$ 의 의미가 같을 경우  $same(x, y)$ 라 표현한다. 메타데이터 스키마  $S$ 의 데이터 요소  $a$ 를 지정하여 의미를 추출할 때,  $get\_element(S, a)$ 라 표현한다.  $set\_element(S(x), a)$ 는 데이터 요소  $a$ 를 메타데이터 스키마  $S$ 의 데이터 요소  $x$ 에 사상시키는 것을 의미한다.  $x$ 와  $y$ 를 합성하여  $z$ 를 생성하는 경우  $z=+(x,y)$ 라 표현한다. 표 4는 구조 불일치 해결을 위한 네가지 사상 규칙을 정의한 것이다.

#### 4. MSDL과 메시지 변환 시스템

정보 교환 시스템이 메타데이터의 변화에 종속되지 않도록 하기 위해서는 메타데이터간의 사상 정보를 사상 규칙과 분리하여 관리하여야 한다. 이 장에서는 메타데이터간의 사상 정보를 독립적으로 관리하기 위하여, XML(eXtensible Markup Language)[11] 기반의 메타데이터 의미 표현 언어인 MSDL을 정의한다. 그리고 MSDL을 이용한 메시지 변환 시스템의 구조와 메시지 변환 프로세스에 대하여 기술한다.

##### 4.1 MSDL

MSDL은 상이한 두 메타데이터 간의 의미 전달에 필요한 의미, 표현, 및 구조상의 차이점을 기술하기 위해 정의된 XML기반의 언어로서 그림 2와 같은 구조를 지닌다. MSDL 메시지는 <MSDL>이라는 하나의 루트 요소를 가진다. MSDL이 담고 있는 정보는 크게 세 가지 구성 요소로 나누어 볼 수 있다. 구성 요소로는 대상 데이터 요소들을 유일하게 표현하기 위한 네임스페이스 부분, 데이터 요소간의 사상관계를 표현하는 <MAP>부분, 사용하는 코드셋의 사상을 표현하는 <CodeSet>부

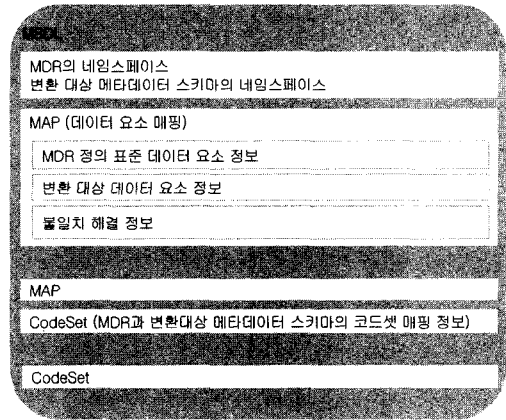


그림 2 MSDL의 구조

분이 있다.

네임스페이스 부분은 MDR의 표준 데이터 요소와 변환 대상 메타데이터 스키마의 데이터 요소들이 사용하는 네임스페이스를 기술한다. 이를 통해 다른 메타데이터 스키마에서 사용하는 데이터 요소들과의 중복을 방지한다.

<MAP>은 데이터 요소간의 의미 전달을 위한 의미, 표현, 구조적 차이점 정보를 지닌다. <MAP>은 하나의 의미를 전달하는데 요구되는 데이터 요소간의 사상을 나타내는 기본 단위이다. MSDL은 다수의 <MAP>을 통해서 두 개의 상이한 이질 메타데이터 스키마의 전체 데이터 요소에 대한 사상 관계를 나타낸다. <MAP>이 담고 있는 정보는 MDR의 표준 데이터 요소 정보, 변환 대상 메타데이터 스키마의 데이터 요소 정보, 불일치 정

표 4 구조 불일치 해결을 위한 사상 규칙

<p>Function 1. 대체(Substitution)를 위한 사상 규칙</p> <pre> set_element(MDR(y), get_element(schema.x)) ⇐ same(get_element(MDR, y), get_element(schema, x))                     </pre>
<p>Function 2. 합성(Composition)을 위한 사상 규칙</p> <pre> set_element(MDR(z), +(get_element(schema, x), get_element(schema, y))) ⇐ same(get_element(MDR, z), +(get_element(schema, x), get_element(schema, y)))                     </pre>
<p>Function 3. 분해(Decomposition)를 위한 사상 규칙</p> <pre> set_element(MDR(x'), x) and set_element(MDR(y'), y) ⇐ same(get_element(schema, z), +(get_element(MDR, x'), get_element(MDR, y')))                     </pre>
<p>Function 4. 재배치(Rearrangement)를 위한 사상 규칙</p> <pre> set_element(MDR(x), y') and set_element(MDR(y), x') ⇐ same(get_element(MDR, x), get_element(schema, y')) and same(get_element(MDR, y), get_element(schema, x'))                     </pre>

표 5 MSDL 스키마 (MSDL Schema)

```

.....
<xs:element name="msdl">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="mdrNamespace" type="xs:string"/>
      <xs:element ref="localNamespace"/>
      <xs:element ref="map" maxOccurs="unbounded"/>
      <xs:element ref="CodeSet" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
.....
<xs:element name="map">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="mdrElement" type="elementIDType" maxOccurs="unbounded"/>
      <xs:element name="localElement" type="elementPathType" maxOccurs="unbounded"/>
      <xs:element ref="mappingRule"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
.....
<xs:simpleType name="mappingType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="substitution"/>
    <xs:enumeration value="composition"/>
    <xs:enumeration value="decomposition"/>
  </xs:restriction>
</xs:simpleType>
.....

```

보의 세 가지가 있다. 표준 데이터 요소와 변환 대상 데이터 요소는 각각 데이터 요소 이름, 데이터 요소의 식별을 위한 식별자, 데이터 형, 그리고 필요에 따라 데이터 요소의 측정단위를 기술한다. <CodeSet>에는 <MAP>에 기술한 데이터 요소의 데이터 타입이 코드셋 타입인 경우, 이에 해당하는 코드셋의 이름과 목록을 사상하고자 하는 순서대로 나열한다. 표 5는 MSDL을

정의한 XML 스키마의 일부이다.

4.2 XML 메시지 변환 시스템의 구조

이 시스템은 XML 메시지의 데이터를 사용자가 인식할 수 있는 형태로 자동 변환 해준다. 시스템의 주요 부분은 MDR, MSDL 레지스트리(MSDL Registry), 사용자 인터페이스, 그리고 메시지 변환기(Message Converter)의 네 부분으로 구성된다. 그림 3은 XML 메시

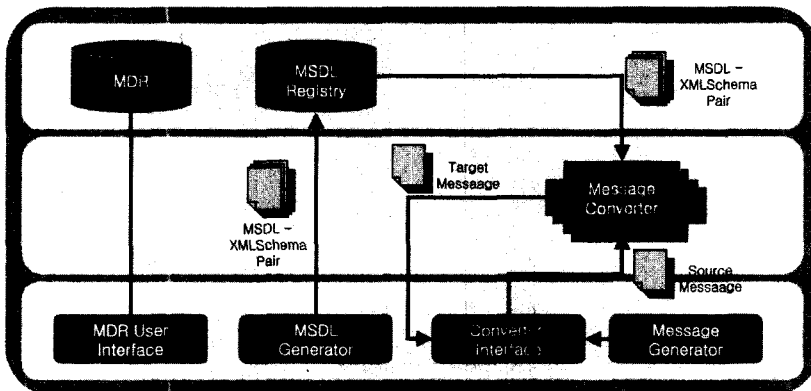


그림 3 XML 메시지 변환 시스템의 구성

지 변환 시스템의 전체적인 구조를 보여준다.

MDR은 ISO/IEC 11179 스펙에 따라 설계되며, 변환의 기준점이 되는 표준 데이터 요소를 등록·관리한다. MDR의 표준 데이터 요소는 표준화 프로세스에 의해 변화될 수 있으므로 특정 시점의 데이터 요소를 XML 메시지로 배포하고, 배포한 XML 메시지를 MDR에서 관리한다. 사용자는 MSDL 작성시 특정 시점에 배포된 MDR의 데이터요소를 사용하고, 사용하는 버전을 명기함으로써 표준 데이터 요소의 변화로 인한 혼란을 방지한다[12].

MSDL 레지스트리는 표준 데이터 요소와 변환 대상 메타데이터의 데이터 요소의 사상 정보를 가지고 있는 MSDL을 관리하고 배포하는 역할을 한다. MSDL 레지스트리는 MSDL과 관련된 XML 스키마를 함께 관리한다. 메타데이터 스키마에 변화가 생길 경우, 사용자는 수정된 MSDL과 XML 스키마를 MSDL 레지스트리에 재등록함으로써 메타데이터 스키마의 변화를 반영할 수 있다.

사용자 인터페이스는 사용되는 목적에 따라 네 가지로 구분된다. 인터페이스는 표준 데이터 요소 검색을 위한 MDR 사용자 인터페이스(MDR User Interface), MSDL 문서 생성을 위한 MSDL 생성기(MSDL Generator), 자신이 가지고 있는 데이터를 XML 메시지 형태로 만들기 위한 메시지 생성기(Message Generator), 그리고 메시지 변환기와 데이터를 주고받기 위한 메시지 변환기 인터페이스(Converter Interface)가 있다.

메시지 변환기(Message Converter)는 표현, 및 구조 불일치 해결을 위한 사상 규칙을 지닌다. 메시지 변환기는 입력받은 소스메시지를 타깃 메시지로 변환하며, 생성된 타깃 메시지를 사용자에게 반환한다. 상세한 메시

지 변환 프로세스에 대해서는 다음 절에서 기술한다.

### 4.3 XML 메시지 변환 프로세스

메시지 변환은 다섯 가지의 정보를 입력받아 이를 이용하여 수행된다. 다섯 가지의 정보는 소스 메시지, 소스 메시지의 XML스키마, 소스 메시지의 MSDL, 타깃 메시지의 XML 스키마, 타깃 메시지의 MSDL이다.

메시지 변환기는 불일치 해결을 위해 2단계로 나누어 사상 규칙을 적용한다. 그림 4는 이러한 메시지 변환 프로세스를 도식화한 것이다. 1단계에서는 소스 메시지를 표준 데이터 요소로 구성된 표준 메시지로 변환한다. 메시지 변환기는 사용자가 메시지 변환기 인터페이스를 통해서 입력한 소스 메시지를 받는다. 그리고 변환에 필요한 소스 메시지의 MSDL과 XML 스키마 쌍을 그림 3에 나타난 MSDL 레지스트리에 요청하여 이를 전달받는다.

메시지 변환기는 MSDL의 각<MAP>에 기술된 데이터 요소를 사상 시킨다. 구조 불일치의 경우 <MAP>에 기록된 사상 규칙의 이름에 따라 3.3절에서 설명한 사상 규칙을 적용하여 변환한다. 표현 불일치는 구조 불일치 사상을 완료된 후, 3.2절의 사상 규칙을 적용하여 <MAP>지정된 표준 데이터 요소의 형식으로 데이터를 변환하여 해결한다. 각 <MAP>에 있는 데이터 요소별 사상이 끝나면 표준 메시지가 생성된다. 2단계에서는 1 단계에서 생성된 표준 메시지를 사용자가 요청한 타깃 메시지로 변환한다. 2단계는 소스메시지 대신 표준 메시지가 이용되고, 사상 규칙 적용시 합성과 분해를 역으로 적용시키는 것만 다르고 나머지는 1단계와 유사 형태로 진행된다. 2단계까지의 변환이 끝나면 생성된 타깃 메시지를 그림 4에 나타난 사용자 인터페이스로 반환한다. 메시지 변환 프로세스는 이러한 두 단계의 과정을 통해

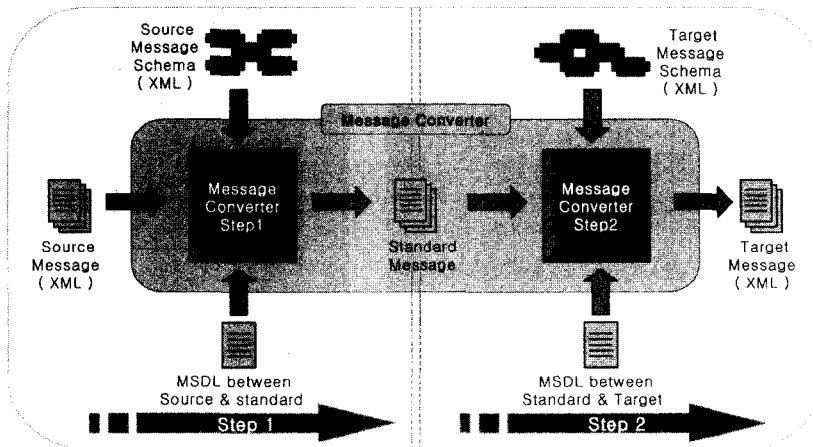


그림 4 XML 메시지 변환 프로세스

서 복합적인 구조 불일치를 해결하기 위한 사상규칙의 조합 효과를 얻을 수 있으며, 복합적인 불일치를 해결할 수 있다.

4.4 XML 메시지 변환 예제

이 절에서는 MSDL을 이용한 간단한 구조 불일치 해결의 예를 소스와 함께 설명한다. 예제를 기술하기 위하여 다음과 같은 상황을 가정한다.

A 기업은 B 출판사로부터 서적을 인수 받아 판매하는 기업이다. B 기업이 새로 출간된 서적 정보를 받기 위해, MDR의 데이터 표준을 기준으로 상호간의 데이터를 교환하기로 합의를 하고 서로가 사용하는 XML 문서에 대한 MSDL을 각각 작성하였다. 편의상 기업 A의 메시지를 메시지-A, MDR의 표준 데이터 요소로 변환 메시지를 표준 메시지, 그리고 기업 B의 메시지 형태로 변환된 메시지를 메시지-B라고 하자. 이 메시지들은 서적의 제목과 저자 성명, 가격 정보를 가지고 있다.

메시지-A, 표준메시지, 메시지-B의 XML 스키마가 다음의 그림 5와 같은 경우, 메시지들간의 변환시에 구조 불일치가 발생한다.

이 경우 대부분은 기본 사상 규칙인 대체(Substitution) 연산을 통해 메시지를 변환할 수 있다. 그러나 저자이름을 전달하기 위해서는 합성 연산을 통해서 FirstName과 LastName을 결합하여만 표준 메시지의 AuthorName을 나타낼 수 있다. 표 6은 이 예제에서 사용되는 데이터 요소간 사상 정보를 기술한 MSDL 문서의 일부를 보여준다.

메시지-A의 MSDL에서 합성(Composition) 규칙을 이용해서 저자 이름을 전달하는 부분은 다음 표 5와 같이 기술 할 수 있다. 대체(Substitution)를 통해 사상되는 다른 데이터 요소들은 <mappingType>을 Substitution으로 기술한다. 다른 부분을 기술하는 방식은 이와 유사하게 이루어지므로 생략한다.

메시지-A의 인스턴스가 그림 5의 (a)와 같다면, 메시지-A의 MSDL을 이용하여 XML 메시지 변환의 1단계

표 6 메시지-A의 MSDL

```

<msdl>
.....
<map>
  <mdrElement>
    <elementName>AuthorName</elementName>
    <elementID>DE008201</elementID>
    <dataType>
      <nonCodeSetDataType>string</nonCodeSetDataType>
    </dataType>
    <measurementUnitID>MU000000</measurementUnitID>
  </mdrElement>
  <localElement>
    <elementName>FirstName</elementName>
    <elementPath>/Product/Book/FirstName</elementPath>
    <dataType>
      <nonCodeSetDataType>string</nonCodeSetDataType>
    </dataType>
    <measurementUnitID>MU000000</measurementUnitID>
  </localElement>
  <localElement>
    <elementName>LastName</elementName>
    <elementPath>/Product/Book/LastName</elementPath>
    <dataType>
      <nonCodeSetDataType>string</nonCodeSetDataType>
    </dataType>
    <measurementUnitID>MU000000</measurementUnitID>
  </localElement>
  <mappingRule>
    <mappingType>composition</mappingType>
    <delimiter>,</delimiter>
  </mappingRule>
</map>
.....
</msdl>
    
```

가 수행된다. 그림 5의 (b)는 1단계 변환 수행의 결과물인 표준 메시지이다. 이 표준 메시지는 메시지 변환 2단

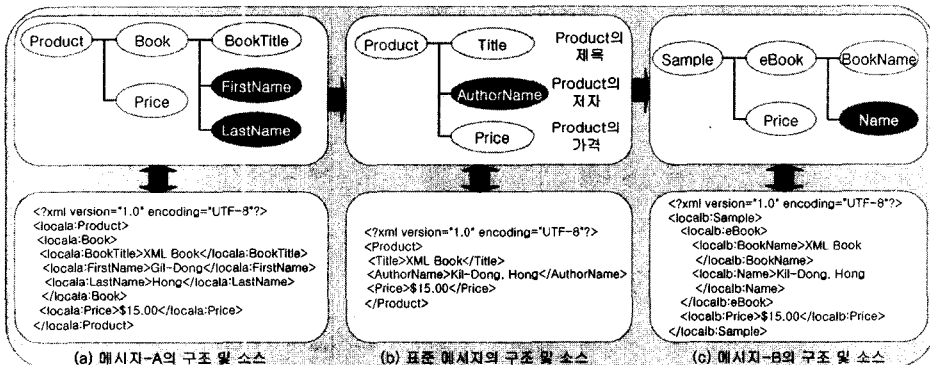


그림 5 메시지-A, 표준 메시지, 메시지-B의 구조 및 소스

계의 수행완료 시기까지만 임시로 관리된다.

XML 메시지 변환은 그림 5의 (b)의 표준 메시지를 이용하여, 변환의 2단계를 수행함으로써 완료된다. 2단계 변환을 위해서는 기업-B가 작성한 MSDL이 필요하다. 그림 5의 (c)는 메시지 변환의 최종 결과물인 메시지-B이다.

## 5. 구현 및 비교 평가

### 5.1 구현

XML 메시지 변환 시스템에서 요구되는 서비스인 표준 데이터 요소 검색, 메시지 생성, 메시지 변환에 대해 시스템을 구현하였다. 개발언어는 주로 ASP.NET을 이용하였으며, 운영체제는 Windows 2000을 사용하였다. DBMS는 ASP.NET과의 호환성을 위해서 SQL Server 2000을 사용하였다.

MDR의 표준 데이터 요소와 표준 데이터 요소를 표현하는데 사용되는 기본적인 데이터형과 코드셋에 관한 정보는 XML 문서 형태로 제공된다. 사용자는 검색용 브라우저를 이용하거나 MDR의 표준 데이터 요소에 대한 정보를 지닌 XML 메시지를 직접 판독하여 표준 데이터 요소에 대한 검색 및 정보를 얻을 수 있다. 사용자는 표준 데이터 요소와 변환 대상이 되는 자신의 데이터 요소를 비교한 뒤 MSDL 생성기를 이용하여, 이들 간의 사상을 만든다. 만들어진 MSDL은 MSDL 레지스트리에 등록된다. 이 때, 사용자는 MSDL을 작성하는데 메타데이터 스키마의 XML 스키마를 함께 등록한다.

사용자가 소스 메시지를 입력하면서 타깃 메시지의 형태를 알려주면, 메시지 변환기는 MSDL 레지스트리

에 관련된 MSDL과 XML스키마를 요청한다. 작업에 필요한 정보가 모두 갖춰지면, 앞 절에서 설명한 두 단계를 거쳐 메시지를 변환한다. 최종 변환 형태인 타깃 메시지가 생성되면 타깃 문서의 XML스키마를 이용하여 메시지의 유효성을 검사하고, 메시지의 유효성이 판별되면 사용자에게 생성된 타깃 메시지를 반환함으로써 모든 변환 절차를 마친다. 그림 6은 구현된 메시지 변환기의 사용자 인터페이스 화면이다.

Load Source Message 버튼을 이용하여 소스 메시지를 지정한 뒤, 타깃 메시지의 형태를 선택한다. 타깃 메시지는 ONIX[14], Dublin Core[15], 그리고 임의로 작성한 Local Standard의 메타데이터 스키마 중에서 한 가지를 선택할 수 있다. 타깃 메시지를 선택하면, 변환기는 타깃 메시지를 생성하여 화면의 오른쪽 하단에 보여준다. 사용자는 Status Information을 통해서 변환의 성공 여부를 판단할 수 있다. Save Target Message 버튼을 눌러 메시지의 저장을 시도하면, 새로 생성된 타깃 메시지의 유효성을 검증하여 유효한 메시지임이 판별되면 저장한다.

### 5.2 비교 평가

구현된 메시지 변환기의 성능을 평가하기 위해서 메타데이터를 통한 데이터 공유 및 교환에 관한 연구가 가장 활발히 진행되고 있는 서지정보 분야를 대상으로 하였다. 서지 정보 분야에서는 ONIX, Dublin Core 및 MARC[16] 등 다양한 표준 메타데이터 스키마들이 사용되고 있으며, 또한 이를 따르지 않는 비표준 메타데이터 스키마들도 일부 존재한다.

이 논문에서는 ONIX, Dublin Core, 그리고 임의로

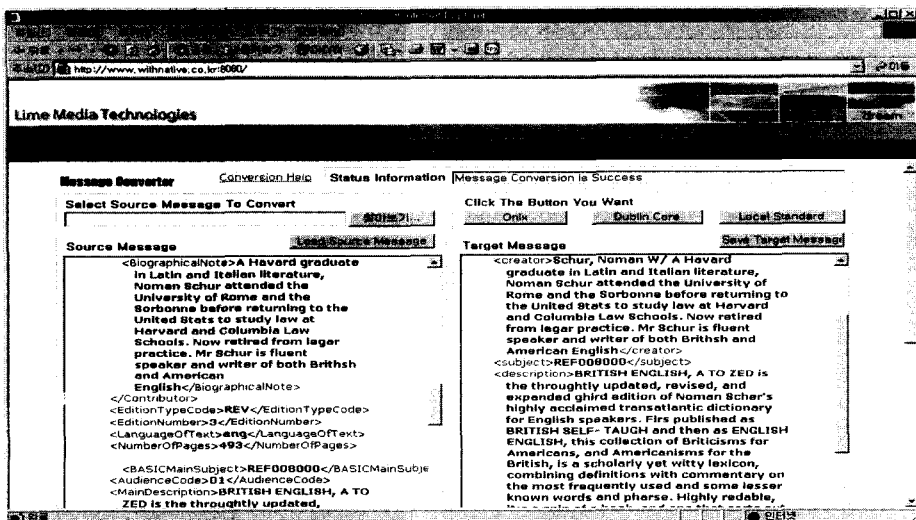


그림 6 XML 메시지 변환기 구현 화면



표 7 MSDL 기반 XML 메시지 변환 시스템의 실험 결과

구분	표준		비표준
메타데이터 스키마 종류	ONIX	Dublin Core	Random schema
데이터 요소 개수	44	9	7
다른 종류의 메타데이터 스키마와의 사상 관계 (다른 종류와의 동일 의미 데이터 요소 개수)	DC(9) Random schema(7)	ONIX(9) Random schema(6)	ONIX(7) Random schema(6)
변환 결과 (변환된 데이터 요소 개수)	DC(9) Random schema(7)	ONIX (9) Random schema(6)	ONIX (7) Random schema(6)

선정한 메타데이터 스키마(Random schema)의 인스턴스들을 대상으로 선정하여 XML 메시지 변환 시스템에 적용하였다. 샘플에 사용하는 데이터 요소의 선정은 ONIX와 Dublin Core의 인스턴스 생성시 빈번하게 사용되는 데이터 요소와 ONIX 스펙에서 제공하는 샘플을 기준으로 하였다. 각 메타데이터 스키마에 해당하는 MSDL을 작성한 뒤, 각 메타데이터 스키마의 인스턴스를 생성하여 변환을 수행하였다. 표 7은 실험 결과를 보여준다.

위 실험 결과에서, 동일한 의미를 가지는 데이터 요소들간의 데이터 변환 및 데이터 의미전달이 구현된 시스템을 통해 성공적으로 이루어졌음을 할 수 있다. 그러나 목적으로 하는 적절한 형태의 메타데이터 스키마가 존재하지 않는 데이터 요소의 경우 데이터의 유실이 발생한다. 수신자 측에서 인식할 필요가 없는 데이터 요소로 인한 데이터의 유실은 손실이라 볼 수 없으므로 성공적으로 변환되었다고 간주할 수 있다.

이 논문에서 제안한 XML 변환 시스템은 메타데이터 스키마 변화에 대한 시스템 유연성과 표준 데이터 요소의 사용으로 인한 표준화 기여, 그리고 다자간 데이터 교환에 제약이 없다는 장점을 지닌다. 이를 요약하면 다음과 같다.

- 첫째, 변환 대상 데이터 요소를 MDR에서 제공하는 표준 데이터 요소와 사상한다. 따라서 데이터 수신자에 대한 고려 없이도 메시지의 교환 및 변환이 가능하다.
- 둘째, 데이터 요소간의 사상 관계와 시스템 구현을 분리한 개념의 적용하므로 메타데이터 스키마 변화로 인한 시스템 유지 보수 비용 및 시간이 적게 든다.
- 셋째, MSDL의 작성을 통해 다자간 데이터 교환이 가능하며, 데이터를 교환하는 대상의 변화와 그 수의 변화에 따른 제약이 없다.

메타데이터를 이용한 데이터 교환을 위한 기존 시스템으로는 David wang이 제안한 X-MAP[13] 시스템과 상용화된 BizTalk[7]이 있다. X-MAP 시스템은 XML로 데이터를 교환하는 다중의 이질 시스템간의 상호운용성 조정을 위해 스키마 요소간의 의미를 연관시켜주는 반자동 시스템이다. Microsoft가 상용화한 BizTalk

는 BizTalk Mapper를 이용하여 스키마간의 사상을 직접 만들어준다.

이 논문에서 제안하는 메시지 변환 시스템은 메타데이터 스키마의 변형시 해당 메타데이터 스키마를 반영하는 MSDL의 수정만이 요구되며, 시스템 전체에 대한 수정 및 재구성이 요구되지 않는다. 그 결과 기존 시스템에 비해서, 시스템을 유지 보수하기 위한 노력과 시간을 단축할 수 있는 장점을 가진다. 표 8은 기존 시스템들과 이 논문에서 제안한 시스템을 비교·요약한 내용이다.

표 8 기존 시스템과 MSDL기반 XML 변환 시스템의 비교

	MSDL 기반 변환 시스템	X-MAP	BizTalk
표준성 (ISO/IEC 11179)	○	X	X
자동화	○	△	○
XML 지원	○	○	○
시스템 유연성	○	△	X
다자간 메시지 교환	○	X	○

○ : 지원, △ : 일부 지원, X : 지원하지 않음

표 8에서 표준성은 표준화된 데이터 요소의 사용을 통한 표준화 기여 여부를 의미한다. 표준 데이터 요소의 사용은 향후 생성될 데이터 요소가 참고할 표준의 적용을 의미하므로, 추가적인 데이터 불일치 발생을 방지한다. 자동화는 데이터 요소간의 사상이 끝난 이후에 데이터 전달을 위한 과정에 대한 자동화 여부를 의미한다. 이 외에도 MSDL기반 XML 메시지 변환 시스템은 데이터 전달을 위한 표준인 XML을 지원하며, 메타데이터 스키마 변화에 따른 시스템의 유연성이 뛰어나며, 다자간 메시지 교환을 지원한다.

### 6. 결론

기존의 메타데이터 게이트웨이는 시스템이 메타데이터 스키마에 종속되어 있어 메타데이터 스키마의 변화에 따른 시스템의 유지보수에 많은 비용과 시간이 소요된다. 이러한 기존의 스키마 종속적인 시스템의 단점을 보완하기 위해서는 이질적인 메타데이터간의 불일치를

해결하기 위한 사상 정보와 사상 규칙을 분리한 시스템을 개발해야 한다.

이 논문에서는 MSDL(Metadata Semantics Description Language)을 정의함으로써 이질적인 메타데이터간의 사상정보를 시스템에 독립적으로 관리할 수 있도록 하였다. 이 논문에서 제안한 시스템은 MSDL을 통해 사상 정보를 독립적으로 관리하므로 기존 시스템에 비해서 메타데이터 스키마의 변화에 따른 시스템의 유지보수가 용이하다. 또한 MSDL이 작성된 임의의 메타데이터 스키마에 대해서 데이터 교환 대상의 변화와 수에 따른 제약이 없이 다자간 변환이 가능하다.

MSDL에 메타데이터간의 사상 관계를 기술할 때는 ISO/IEC 11179에서 제안하는 MDR의 표준 데이터 요소를 기준으로 사용한다. 메타데이터 레지스트리를 통한 표준화된 데이터 요소의 제공은 향후 생성될 데이터들의 기준점으로 사용될 수 있다. 따라서 메타데이터 표준화를 통한 메타데이터 불일치 발생을 방지하는 효과를 얻을 수 있다.

이 논문에서 제안한 시스템의 구축을 통해 MDR을 이용한 데이터 교환 시스템의 장점과 응용성에 대해 알 수 있었다. 그러나 MDR의 표준 데이터 요소의 실제적인 등록 및 관리를 위한 부분은 이 논문의 주제를 벗어나므로 포함시키지 않았다. 향후 연구로는 이 논문에서 제안한 시스템의 실제적 적용을 위해, ISO/IEC 11179 MDR의 관리를 위한 케이스 도구의 개발이 요구된다. 또한 제안하는 시스템을 전자상거래 및 지식 정보 교환에 이용하기 위해 기존의 전자상거래 프레임워크와의 연동 및 다양한 분야로의 응용을 위한 연구가 필요하다.

**참 고 문 헌**

[1] 백두권, "정보통신 및 표준화 기술 동향: 데이터 표준화와 메타데이터 레지스트리", TTA 저널 통권71호, 2000.

[2] C. Bianchi, J. Petrone, "Distributed Interoperable Metadata Registry," D-Lib Magazine, December 2001.

[3] G. Wiederhold, "Intelligent Integration of Information," ACM SIGMOD international conference on Management of data, Vol. 22, June 1993.

[4] W.Kim and J.Seo, "Classifying schematic and data heterogeneity in multi-database systems," IEEE Computer, Vol.24, No.12, pp.12-18, 1991.

[5] C. Parent, S. Spaccapietra, "Issues and approaches of database integration," Communications of the ACM, May 1998.

[6] R. D. Holowczak and W.-S. Li, "A Survey on Attribute Correspondence and Heterogeneity Metadata Representation," FIRST IEEE METADATA CONFERENCE, April 16-18, 1996.

[7] Microsoft BizTalk, <http://www.microsoft.com/biztalk/>

[8] ISO/IEC IS 11179, "Information technology - Specification and standardization of data elements," 2003.

[9] 김진관, 김종일, 최오훈, 백두권, "MDR을 이용한 XML DTD 이질성 해결기법", 한국정보과학회 추계학술발표 논문집(1권), pp 67-69, 2002.

[10] Knowledge Interchange Format (KIF), <http://logic.stanford.edu/kif/kif.html>

[11] Extensible Markup Language (XMLTM) <http://www.w3.org/XML/>

[12] 김진관, 김종일, 백두권, "MDR기반 이질 스키마간 XML문서 자동 변환 기법", 정보처리학회 춘계학술발표 논문집(중), pp1449-1452, 2003.

[13] D. Wang, "Automated Semantic Correlation between Multiple Schema for Information Exchange," M.I.T., MM, May 2000.

[14] ONIX(ONline Information eXchange) International, <http://www.editeur.org/>

[15] Dublin Core Metadata Initiative, <http://dublin-core.org>

[16] MARC, <http://www.loc.gov/marc/>



**김진관**  
 2002년 고려대학교 컴퓨터학과(학사)  
 2004년 고려대학교 컴퓨터학과(석사)  
 관심분야는 데이터베이스, 메타데이터 레지스트리, 웹 서비스, 유비쿼터스



**김종일**  
 1992년~1997년 고려대학교 원예학과  
 1998년~1999년 고려대학교 컴퓨터학과(석사). 1998년~1999년 고려대학교 컴퓨터과학기술연구소 연구원. 2000년~현재 (주)라임미디어테크놀러지스 기술이사. 관심분야는 메타데이터, 워크플로우, 원격 교육시스템



## 정 동 원

1997년 군산대학교 전산과(학사). 1998년 한국전자통신연구원(위촉연구원). 1999년 충북대학교 전산과(석사). 1999년~2000년 ICU 부설 한국정보통신교육원(전임강사). 2001년~현재 (주)라임미디어 테크놀로지(연구원). 2002년~현재 TTA 표준화위원회(특별위원). 2004년 고려대학교 컴퓨터학과(박사) 2004년~현재 고려대학교 컴퓨터학과(연구교수). 관심분야는 분산 컴퓨팅, 이동 에이전트, 데이터베이스, 메타데이터 레지스트리, 유비쿼터스



## 백 두 권

1974년 고려대학교 수학과 학사. 1976년 고려대학교 대학원 산업공학과 석사 1983년 Wayne State Univ. 전산학 석사. 1986년 Wayne State Univ. 전산학 박사. 1986년~현재 고려대학교 컴퓨터학과 교수. 1989년~현재 한국정보과학회 이사/평의원. 1991년~현재 ISO/IEC JTC1/SC32 국내위원회 위원장. 1992년~현재 한국시물레이션학회 이사/부회장/회장. 2002년~2003년 고려대학교 정보통신대학 학장. 2002년 ~현재 과학기술정보표준위원회 위원장. 관심분야는 정보보호, 데이터베이스, 소프트웨어공학, 데이터공학, 컴포넌트 기반 시스템, 메타데이터 레지스트리, 정보통합