
웹 환경에서 데이터 상호운용을 위한 XMDR 기반의 검색 시스템 설계

문석재* · 정계동* · 최영근*

Design of Retrieval System based on XMDR for Data Interoperability in a Web Environment

Seok-Jae Moon* · Gye-Dong Jung* · Young-Keun Choi*

요 약

최근 기업들은 레거시 시스템들간의 데이터 상호 운용하기 위해 기업들은 이미 보유하고 있는 레거시 업무와 EAI 시스템을 도입하고 있다. 협업적인 거래 환경에서의 EAI 시스템은 유기적으로 통합하고 공유함으로써 효율적인 검색을 기대할 수 있다. 그러나 기존 레거시 시스템은 특정 목적에 따라 설계단계부터 상호 운용성을 고려하지 않고 독자적으로 관리되므로 EAI는 표준기술 적용이 어려우므로 별도의 전용 EAI 솔루션을 도입해야 한다. 이러한 문제를 해결하기 위해 데이터 통합을 이용하여 메타데이터 레지스트리를 이용한다. 그러나 메타데이터의 다양한 타입과 의미론적 명세, 데이터 이질성 문제, 이기종간의 시스템 이질성에 대한 문제도 야기된다. 따라서 본 논문에서는 웹 환경에서 데이터 상호운용을 위한 XMDR(eXtended Meta-Data Registry)기반의 검색 시스템을 제안한다.

ABSTRACT

Recently enterprises introduce EAI systems and legacy business which already obtained for data integration among legacy systems. EAI systems in cooperative transaction environment can be expected efficient retrieval as sharing and integrating. However existing legacy systems have to introduce particular EAI solution because it is difficult to adjust standard technology to EAI due to be managed independently without considering interoperability. For solving these problems we use metadata registry using data integration. Various types, semantic specification, data heterogeneity and heterogeneity of systems, however, are occurred. Therefore retrieval system based on XMDR(eXtended Meta-Data Registry) for data interoperability in the web environment are proposed in this paper.

키워드

XMDR(eXtended Meta-Data Registry), Ontology, EAI(Enterprise Application Integration), Distributed System

I. 서 론

현재 각 기업들의 필요에 의한 시스템간의 통합은 상호간에 일대일 형식으로 사안에 따라 이루어지고 있다. 이러한 방식의 통합은 레거시 시스템의 종류 및 수가 증

가함에 따라 적용 기술 또한 더 복잡해졌다. 이러한 문제들을 해결하기 위해 도입한 EAI는 기존 일대일 방식과는 달리 새로운 시스템을 도입할 때 어댑터나 커넥터만 추가해 주면 기존에 구축되어있는 시스템들과 손쉽게 통합할 수 있게 되었다[1]. 그러나 EAI는 표준기술을 적용하

기가 어려워 각 프로젝트의 필요에 맞는 별도의 전용 EAI 솔루션을 도입해야하는 문제가 따른다. 또한 기존 레거시 시스템은 특정 목적에 따라 설계 단계부터 상호 운용성을 고려하지 않았고, 독자적으로 관리되고 공유되므로 사실상 통합하는데 어려움이 있다[2]. 이러한 문제를 해결하기 위한 방안으로 MDR(MataData Registry) 표준이 제시되었지만, 다양한 형태의 데이터 타입과 데이터의 의미론적 명세에 대한 이질적인 문제는 여전히 남아있다[3]. 본 논문에서는 웹 환경에서 데이터 상호운용을 위한 XMDR 기반의 검색 프레임워크를 제안한다. 본 검색 시스템은 레거시 시스템간의 통합 있어서 발생하는 데이터의 의미론적 명세에 대한 이질적 문제를 해결하였고, XMDR 질의 템플릿을 사용하여 이기종간의 데이터 처리 및 교환 문제를 해결하였다. 본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구에 대해 서술하고, 제3장에서는 본 논문에서 제안한 XMDR의 정의와 구성을 제4장에서는 XMDR 질의 템플릿에 대해 기술한다. 제5장에서는 XMDR 기반의 검색 시스템 구조의 설계 및 각 모듈에 대해 기술한다. 제6장에서는 시스템 적용 절차를 제7장에서는 시스템 적용과 비교 분석 및 성능 평가에 대해 기술하고, 마지막으로 제8장은 결론 및 향후 연구에 대해서 기술한다.

II. 관련 연구

2.1. 온톨로지(Ontology)

온톨로지는 웹 기반의 지식 처리나 응용 프로그램 사이의 지식 공유, 재사용들을 가능하게 하는 아주 중요한 요소로 자리 잡고 있다. 온톨로지에는 계층분류와 추론규칙에 대한 정의가 포함된다. 계층분류는 객체의 클래스와 서브클래스, 그들 간의 관계를 정의한다. 추론규칙은 프로그램이 새로운 사실을 자동으로 추출하거나 제약조건에 맞지 않는 오류를 찾아내는데 이용된다. 온톨로지를 표현하기 위해 스키마와 구문구조 등을 정의한 언어가 온톨로지 언어이며, 현재 DAML+OIL, OWL, Ontolingua 같은 온톨로지 언어가 정의되었다. 이 중에서 W3C에서 표준안으로 제시한 DAML+OIL은 웹 리소스에 대한 시멘틱 마크업 언어이며 W3C의 RDF와 RDF스키마 표준에 기반을 두고 이들을 확장한 프레임 기반의 온톨로지 표현 언어이다[4].

2.2. 메타데이터 레지스트리(MetaData Registry)

ISO/IEC 11179 MDR에 의해 지속적으로 관리되는 메타데이터들은 정보 시스템의 설계, 구현뿐만 아니라 기존 시스템 간의 데이터 상호운용에 중요한 역할을 할 수 있다. 현재는 각기 다른 기술 구조를 가진 다양한 메타데이터 상호 운용성의 입장에서 통합하기 위한 연구가 이루어지고 있으며, RDF와 같은 시멘틱 웹 기술이 그 예이다. 레거시 시스템들은 다양한 방법으로 데이터베이스를 구축하고 데이터를 표현한다. 이에 따라 각 데이터의 표현 차이와 시스템들 사이에서 이질적 문제가 발생하게 되어 분산된 데이터의 통합을 저해시키고 있다. 이를 해결하기 위한 노력으로, XML을 기반으로 시스템사이의 데이터 교환을 위해 이음동의어를 정의하여 스키마 요소간의 의미를 연관시켜주는 X-MAP 시스템에 대한 제안을 하였고, 이질 데이터 통합 방법으로 질의를 XML형태로 하고 결과를 XML문서로 받는 시스템이 제안되었다[5]. 또한 데이터 통합에 따른 데이터 이질성을 해결하기 위하여, XML기반의 관계형 데이터베이스 메타데이터를 객체지향 데이터베이스에 저장하는 기술과 분산된 데이터의 이질성을 해결하고자 MDR과 온톨로지를 결합하여 데이터를 통합하는 시스템들에 대해 기술하고 있다[6]. 2005년 1월부터 ISO/IEC 11197에서는 XMDR에 대한 프로젝트가 수행되고 있다. 이 XMDR은 데이터 요소들, 용어법, 그리고 MDR을 등록하고 검색하는 것을 더욱 증가시키고, 메타데이터의 다양한 타입들과 의미론적 명세와 질의를 수행할 수 있도록 확대된 능력을 지원하기 위해 MDR 표준인 ISO/IEC 11179 계열의 확장을 목적으로 한다. 이것은 구조화된 메타데이터에 카테고리 분류, 온톨로지, 그리고 시소러스 등의 개념을 결합한 것이다[7].

III. XMDR의 정의와 구성

본 논문에서의 XMDR의 역할은 통합 검색에 필요한 공통 요소를 표준화하여 일관성 있는 검색을 수행할 수 있게 한다. 또한 레거시 시스템마다 처리될 검색 질의문, 질의 결과 등 모든 데이터의 이동은 XML을 이용하여 이질적인 자료구조, 값 그리고 이질적인 컴퓨팅 환경을 일반화시킨다. 본 논문에서 데이터 상호운용을 위한 검색 시스템을 구성하기 위해 제안한 XMDR의 개념적인 구성은 그림1과 같다.

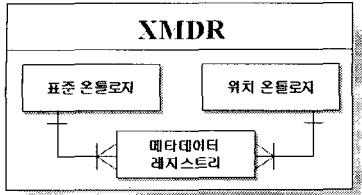


그림 1. XMDR의 구성
Fig. 1. Composition of XMDR

MDR은 레거시 시스템들이 실제 사용되고 있는 데이터베이스의 스키마 정보이고, 표준 온톨로지는 통합 참여되는 레거시 시스템들의 데이터베이스 스키마의 관계성(의미, 표현, 구조 이질성, 메타데이터 제약사항)을 고려하여 메타데이터와 온톨로지를 결합하여 정의하고 표준화 한 것이다. 로케이션 온톨로지는 레거시 시스템들을 접근하기 위한 위치정보, 검색 우선순위 정보를 정의한 것이다. 다음 표들은 XMDR을 구성하기 위한 명세들을 설명한 것이다. 표1은 MDR을 구성하기 위해 정의한 명세이다. 레거시 시스템들에 실질적인 데이터베이스의 메타데이터 스키마 정보를 식별정보, 제약정보, 표현정보로 섹션을 구분하여 정의한다. 표2는 표준 온톨로지를 구성하기 위해 정의한 명세이다. 표준 온톨로지는 참여하는 레거시 시스템의 데이터베이스의 스키마 정보로서 데이터의 의미론적 이질성을 해결하기 위해 공통적인 관계성이 고려된 메타데이터로만 정의하고 표준화 한 것이다. 또한 레거시 시스템의 스키마 정보와 매핑 시 참조되는 사전적인 역할도 한다. 표3은 위치 온톨로지를 구성하기 위해 정의한 명세이다. 레거시 시스템의 데이터베이스에 접근하기 위한 정보를 위치접근 정보, 우선순위 정보로 섹션을 구분하여 정의하였다.

표 1. MDR(MetaData Registry) 명세
Table 1. MDR Specification

섹션	설명	개체
식별 정보	스키마 정보를 고유하게 식별하는 요소	MDR_ID, MDR_DB, MDR_TABLE, MDR_FIELD
제약 정보	스키마 정보의 제약된 사항의 관한 요소	TYPE, SIZE, NOT_NULL, PKEY
표현 정보	스키마 정보의 표현형식에 관한 요소	FORMAT

표 2. 표준 온톨로지 명세
Table 1. Standard Ontology Specification

섹션	설명	개체
표준 메타 데이터 정보	메타데이터의 표준이 정의된 요소	Standard_MetaField
유사어 정보	표준 메타데이터의 유사어 정보	Synonym_MetaField
표준 메타 데이터 제약 정보	스키마 정보의 표현형식에 관한 요소	MetaType, MetaSize, MetaFormat

표 3. 로케이션 온톨로지 명세
Table 3. Location Ontology Specification

섹션	설명	개체
위치 접근 정보	참여 레거시 서버들의 위치 정보 및 접근 권한 정보를 나타내는 요소	MDR_ID, MDR_NAME, LOC_URL, CONN_ID, CONN_PWD
우선 순위 정보	참여 레거시 서버들의 우선순위 가중치와 그 관계를 나타내는 요소	SEARCH_CNT, USER_CNT, PRIORITY

다음 그림2는 위에서 정의한 XMDR의 명세들을 토대로 자산관리 솔루션 레거시 시스템을 통합하는 예로 XMDR의 구성개념 표현한 것이다.

우선 분산된 레거시 시스템의 데이터베이스에 모든 스키마 정보를 수집하여 등록한 것이 MDR이다. 이는 레거시 시스템 관리자가 직접 등록한 정보이다. MDR의 구성은 MDR_ID, MDR_DB, MDR_TABLE, MDR_FIELD, TYPE, SIZE, NOT_NULL, PKEY, FORMAT로 구성된다. 이는 분산된 레거시 시스템의 데이터베이스 실제 사용되는 스키마 정보로 DB명, 테이블 명, 필드 명, 타입, 크기, 널 여부, 주키 여부, 데이터 포맷 등의 메타데이터 정보가 등록된 것이다. 또한 XMDR 구성 및 매핑 하는 과정에서 발생하는 여러 가지 불일치 문제를 해결해야 하는데, 그림2에 예와 같이 '단가' 메타데이터 개체는 각 데이터베이스 스키마의 'Price', 'UNIT_PRICE'으로 스키마 항목이 정의되어 있어 표준 메타데이터 개체와 1:1매핑이 되어 간단하게 처리된다. 반면 '부서' 메타데이터 개체의 경우

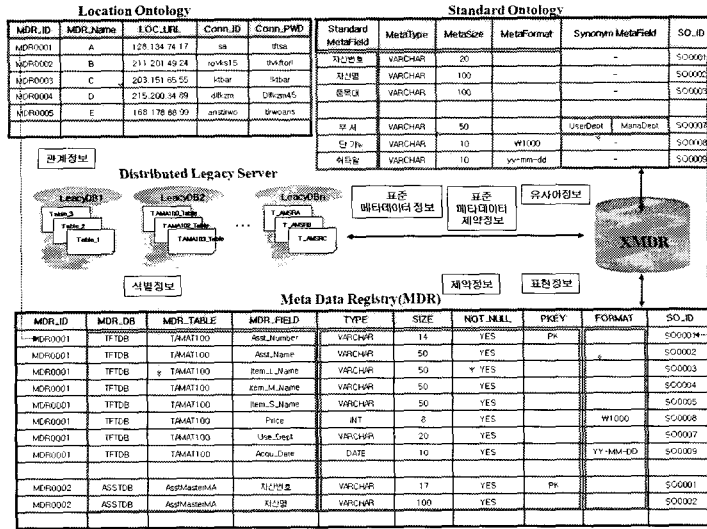


그림 2. XMDR의 구성 개념
Fig. 2. Composition concept of XMDR

는 어떤 데이터베이스 스키마에서는 '사용부서명', 관리 부서명'으로 분리되어 있어 1:N의 관계가 성립한다. 이와 같이 데이터베이스 스키마의 N개의 항목이 표준 메타데이터 개체의 하나의 항목과 매핑 되는 경우는 표준 온톨로지의 유사어 정보에 의해 매핑 된다.

IV. XMDR Query Template

사용자의 검색 요청을 레거시 시스템에서 질의 처리되는 검색 질의문을 구성하기 위해 본 논문에서 정의한 XMDR을 참조하여 XMDR Query 템플릿의 구성을 설명한다. 그림3는 XMDR Query 템플릿의 구성요소를 XML 문서로 정의한 것이다. 그림3과 같이 XMDR Query 템플릿 스키마를 해당 레거시 시스템으로 전송된다.

전송된 템플릿은 레거시 시스템에서 어댑터로 파싱하여 질의로 변환하여 질의문을 처리하게 된다. 이때, 로케이션 온톨로지에 등록된 레거시 시스템의 위치 정보, 접근 정보들을 참조하여 템플릿이 레거시 시스템으로 전송되어 처리된다. 그림4의 XMDR Query 템플릿의 구성 내용은 우선 첫 번째 코드에서는 XML 문서임을 가리키는 <?XML> 요소와 함께 시작한다. 이 주석은 브라우저가 문서를 정확하게 분석하는 것을 보장한다. 두 번째 코드의 root는 루트 요소를 명시하고 네임스페이스를 지정하는

```
<?xml version="1.0" encoding="utf-8" ?>
<XMDR xmlns="http://128.134.59/XMDRTemplate.xsl" >
  <MDR_ID id="000001" >
    <LOCATION >
      <CREATE_TIME><NAME>
      <CONN_ID><CONN_ID>
      <CONN_PWD><CONN_PWD>
      <LOC_LIFE><LOC_LIFE>
      <LOCATION >
      <SQL_QUERY ><SELECT >
      <SEARCH_FIELD id="001" type="varchar" size="14" not_null="yes" pk="yes" format="ss" ><SEARCH_FIELD >
      <SEARCH_FIELD id="002" type="varchar" size="50" not_null="yes" pk="no" format="ss" ><SEARCH_FIELD >
      <SEARCH_FIELD id="003" type="varchar" size="50" not_null="yes" pk="no" format="ss" ><SEARCH_FIELD >
      <SEARCH_FIELD id="004" type="varchar" size="50" not_null="yes" pk="no" format="ss" ><SEARCH_FIELD >
      <SEARCH_FIELD id="005" type="varchar" size="50" not_null="yes" pk="no" format="ss" ><SEARCH_FIELD >
      <SEARCH_FIELD id="008" type="int" size="8" not_null="yes" pk="no" format="Y1000" ><SEARCH_FIELD >
      <SEARCH_FIELD id="007" type="varchar" size="20" not_null="yes" pk="no" format="YY-MM-DD" ><SEARCH_FIELD >
      <FROM ><FROM >
      <MDR_TABLE ><MDR_TABLE >
      <MDR_TABLE ><MDR_TABLE >
      <WHERE >
      <SEARCH_FIELD name="Asst_Number" >
      <VALUE ><VALUE >
      <MDR_FIELD >
      <MDR_FIELD >
      <MDR_FIELD >
      <WHERE >
      <MDR_ID >
      <MDR_ID id="000001" >
      <MDR_ID >
      <MDR_ID >
    </MDR_ID >
  </LOCATION >
</XMDR >
```

그림 3. XMDR Query 템플릿의 스키마 문서
Fig. 3. Schema Document of XMDR Query Template

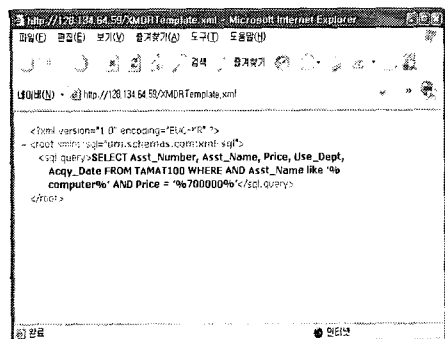


그림 4. XMDR Query Template이 파싱된 결과 화면
Fig. 4. Result of parsed XMDR Query Template

xmlns는 XML 문서에서 요소와 속성 이름으로써 사용할 수 있는 이름 집합을 명시한다. 세 번째 코드는 sql:query 는 각 레거시 시스템별로 생성된 질의문을 명시한다. 이때 질의문은 ANSI 질의문으로 구성되며 레거시 시스템에 실행된다.

V. XMDR 기반의 검색 시스템 구조

본 논문에서 제안한 XMDR 기반의 검색 시스템 구조는 그림5와 같다. 다음은 시스템을 구성하는 통합 프레임워크 계층에서 각 모듈의 기본적인 기능과 설명은 다음과 같다.

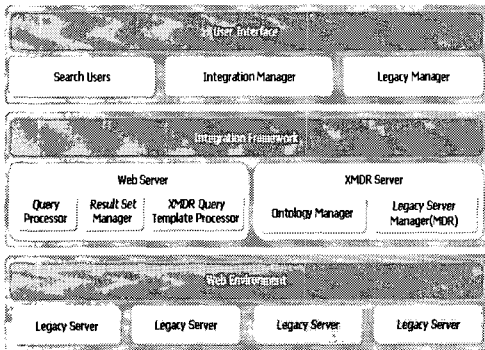


그림 5. XMDR 검색 시스템 구조
Fig. 5. Architecture of XMDR retrieval system

● **질의 처리:** 제안한 시스템은 위의 두 가지 형태의 사용자 질의 처리를 한다. 질의 형태는 기본검색과 확장 검색으로 질의 처리할 수 있게 한다. 사용자가 선택한 시스템 목록 데이터를 참조하여 XMDR에서 레거시 시스템들의 데이터베이스의 검색에 필요한 스키마 정보를 추출한다. 사용자의 질의 데이터와 XMDR에서 추출한 스키마 정보를 추출하여 각 레거시 시스템의 맞는 질의문을 생성한다.

● **XMDR 질의 템플릿 처리:** 질의 처리기에서 넘어온 질의문을 템플릿으로 구성하고 생성되어 사용자가 검색을 원하는 각 레거시 시스템에 전송처리 한다. 질의 처리기에서 넘어온 기본검색 또는 확장검색 질의문과 XMDR 질의 템플릿에서 레거시 시스템별로 생성되어 넘어온 템플릿을 기반으로 각 레거시 시스템에 전송할 XMDR 질의

템플릿을 빌더하게 된다. 빌더 된 템플릿은 각 레거시 시스템에 전송된다.

● **결과 셋 관리:** 각 레거시 시스템마다 검색되어 XML문서로 넘어오면 결과 리시버에서 파서해서 데이터베이스에 저장하게 되고, 결과 처리기는 사용자에게 최종 검색 결과를 보여주게 된다. 또한 레거시 시스템마다 검색 수행시간 및 결과 응답 시간이 다를 수 있어 결과 저장기에서 결과를 수집하여 결과 처리기에 넘기는 딜레이 타임을 버퍼 모듈에서 조절하게 된다.

● **온톨로지 관리:** 온톨로지 관리기의 표준 온톨로지는 통합 참여되는 레거시 시스템들의 데이터베이스스키마의 공통적인 관계성이 고려된 메타데이터로 공통된 부분만을 정의하고 표준화 한 것으로 표준 메타데이터 항목이 등록, 추가, 수정, 삭제가 가능하고 항목이 변경되었다는 것은 사용자의 대표질의 항목이 변경되었다는 것을 의미한다. 로케이션 온톨로지는 레거시 시스템들에 접근하기 위한 메타데이터로 위치정보, 접근권한, 검색 우선순위 정보를 정의하고, 등록한 것이다.

● **레거시 서버 관리:** 레거시 시스템의 정보 변경, 스키마 자체 내용의 내용 및 XMDR 기반의 검색 시스템의 표준 메타데이터와 매핑 내용이 변경 발생할 경우 변경된 내용이 즉시 통합 검색에 반영되도록 한다. 또한 신규로 등록하기 위한 레거시 시스템은 XMDR 기반의 검색 시스템에서 인증되어야만 등록할 수 있으며, 인증여부는 시스템 관리자가 인증한다. 레거시 시스템 관리자는 등록된 시스템의 정보를 수정하거나 삭제할 수 있다. 그리고 서비스 중인 레거시 시스템을 일시 중지할 수 있다. 또한 레거시 시스템 관리기는 현재 통합 서비스에 참여하고 있는 레거시 시스템의 상태를 실시간으로 모니터링 할 수 있다. 이는 분산된 레거시 시스템에 대한 정보를 지속적으로 유지하고, 문서 정보에 대한 연속성, 요소 메타 정보에 대한 일관성을 유지한다. 레거시 시스템의 장애에 대한 신속한 대처를 위해 시스템을 모니터링 하고 검색에 참여시킬 시스템의 목록들을 체크한다.

VI. XMDR 기반의 검색 시스템 적용 절차

그림 6은 사용자가 검색 요청을 하면 XMDR 기반의 검색 시스템에서 분산된 레거시 시스템마다 XMDR Query 템플릿으로 질의 검색을 하여 사용자에게 통합 검색 결과를 보여주는 수행 과정이다. 각 작업을 살펴보면 다음과 같다. 그림7은 통합 참여하는 레거시 시스템들의 표준으로 사용될 메타데이터로 시스템들의 스키마의 공통된 부분만을 표준화한 표준 메타데이터를 XMDR에 등록 및 구성하고, 시스템들의 메타데이터 스키마 정보를 등록하는 과정이다.

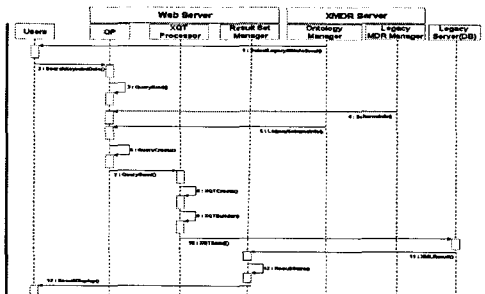


그림 6. 통합 검색 수행 과정
Fig. 6. Process of integrated retrieval

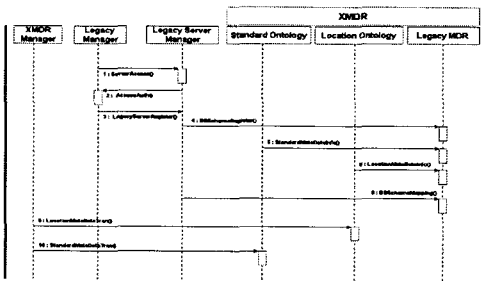


그림 7. XMDR 등록 및 구성 수행 과정
Fig. 7. XMDR registration and composition process

VII. 시스템 적용 및 비교 분석

7.1. 시스템 적용

시스템 적용 사례는 자산 관리 솔루션 레거시 시스템을 통합하는 웹 환경에서 데이터 상호운용을 위한 XMDR 기반의 검색 시스템을 구현하였다. 그림8은 URL, Area, IP, PORT, 데이터베이스 종류, 데이터베이스 명, 언어, 서

비스 여부 등의 레거시 시스템의 정보를 검색 시스템에 등록하는 인터페이스이다. 레거시 시스템 모니터링은 그림 9와 같이 검색 시스템에 참여하고 있는 각 레거시 시스템의 상태를 모니터링 할 수 있다. 이는 분산된 각 레거시 시스템에 대한 정보를 지속적으로 유지하고, 문서 정보에 대한 연속성, 요소 베타 정보에 대한 일관성을 유지한다.

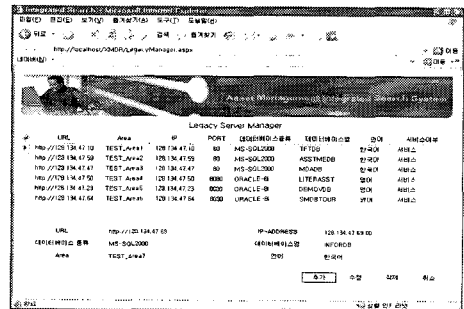


그림 8. 레거시 시스템 정보 등록 인터페이스
Fig. 8. Legacy System information registration interface

시스템 장애에 대한 신속한 대처를 위해 분산시스템을 모니터링하고 통합 검색에 참여시킬 시스템의 목록들을 체크한다.

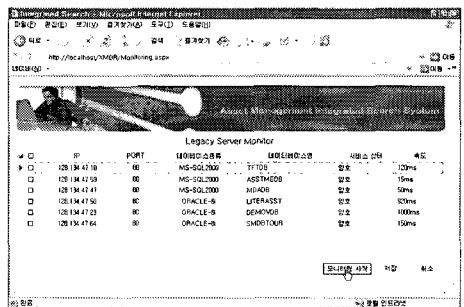


그림 9. 레거시 시스템 모니터링 인터페이스
Fig. 9. Legacy System monitoring interface

그림10은 현재 서비스 중인 레거시 스키마를 선택하면, 각 원천서버에서 이미 매핑된 정보가 있을 경우 매핑 정보를 보여주며, 변경이 필요할 경우 매핑 정보를 수정한다. 현재 통합 검색 프레임워크에 등록된 레거시 시스템의 DB를 선택하고, 테이블, 필드명을 선택한다. 로케이션 온톨로지와 표준 온톨로지에서 현재 매핑할 레거시 시스템의 스키마 정보와 맞는 메타데이터를 선택하여 매핑을 하게 되면 데이터베이스에 즉시 적재된다.

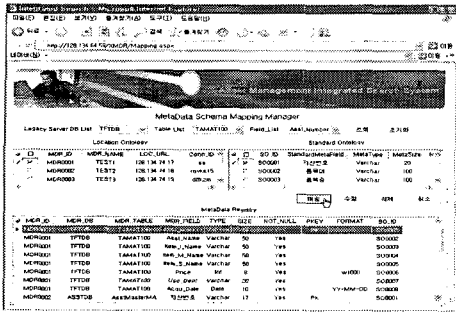


그림 10. 레거시 시스템 스키마 매핑 인터페이스
Fig. 10. Legacy System schema mapping interface

그림11은 검색 시스템에서 사용자에게 제공하는 검색 화면과 검색된 결과이다. 사용자는 검색 분류 선택에서 기본, 확장을 선택한다. 기본검색 선택인 경우에는 포털 검색처럼 검색어를 입력할 수 있게 하였고, 확장 검색을 선택한 경우에는 세부 항목의 검색어를 입력할 수 있게 하였다.

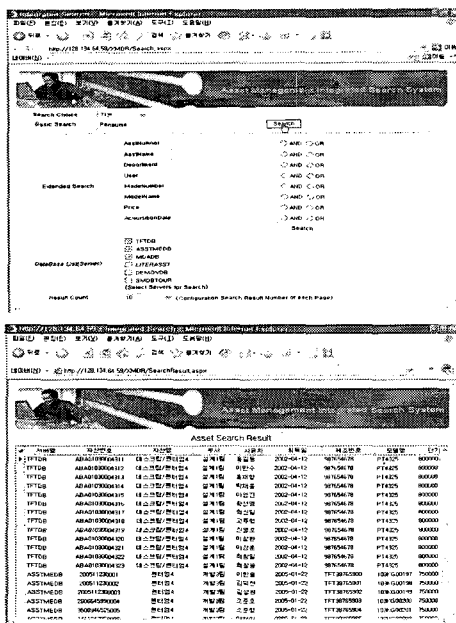


그림 11. 검색 인터페이스 및 검색 결과 화면
Fig. 11. retrieval interface and Result

7.2 비교 분석

본 논문에서는 제안한 시스템은 분산 환경에서 이질성 문제 해결과 데이터 상호운용 및 통합을 지원하기 위한

XMDR 구축에 목적으로 연구된 시스템이다. 따라서 표5에서와 같이 위의 3프레임워크를 대상으로 다음 5가지 항목으로 구분하여 비교하였다. 그 항목으로는 ISO/IEC 11179에서 논의 중인 XMDR 지원여부, 데이터 교환 자동화여부, 범용 명세 스키마 지원여부, 적용 범위, 저장소 구조에 대한 부분으로 비교평가 된다.

표 5. 데이터 통합을 위한 프레임워크들과의 비교
Table. 5 The comparison of frameworks for data integration

비교항목	ETL 기반 프레임워크	db2에 프레임워크	Oracle에 프레임워크	본 프레임워크
XMDR 지원	일부지원	일부지원	일부지원	지원
데이터 교환 자동화	지원	일부지원	일부지원	지원
범용 명세 스키마	지원 없음	지원	지원	지원
적용 범위	모든 분야	모든 분야	IT/EC 분야	모든 분야
저장소 구조	중앙 집중	분산	분산	중앙 집중

7.3. 성능 평가

본 절에서는 XMDR 기반의 검색 정보의 효율성을 알아보기 위해 표준화 되어 있지 않은 자산관리 솔루션 레거시 시스템의 검색 부분과 XMDR을 이용한 검색 부분의 결과를 비교하기 위해 표6과 같은 테스트 환경 및 방법으로 분석한다. 그림12.1과 12.2는 표6과 같은 테스트 환경에서 XMDR과 EAI환경에서 통합 자산 검색을 하는데 소요되는 평균 응답시간을 비교분석하였다.

표 6. 테스트 환경 및 조건
Table. 6 Test environment and condition

테스트 환경	OS : Microsoft Windows Server 2003 CPU : Intel Pentium M2.00GHz RAM : 512MB 사용언어 : ASP.NET, JAVA Script, HTML, MS-SQL, 오라클대
테스트 조건	IT자산 중심검색, 중복 검색 제외 특정자산에 대한 상세 조건 검색(자산번호, 자산명, 사용자, 단가...) 조건에 맞는 검색 평균 응답시간

그림12.1과 같이 웹 환경에서 제공되는 EAI 환경의 정보 검색한 것과 본 논문에서 제안한 XMDR을 이용하여 정보 검색한 것을 비교한 결과 EAI 환경의 정보 검색 평균 응답시간이 현저히 떨어지는 것으로 비교 분석 되었다.

VIII. 결 론

본 논문에서는 레거시 시스템을 통합하는데 발생할 수 있는 시스템의 구성적 문제, 데이터베이스 시스템의 차이, 데이터베이스 구현의 차이, 데이터 표현의 차이와 같은 이질적 문제를 해결하기 위해 레거시 서버의 MDR과 Ontology를 사상시킬 수 있는 방안으로 XMDR을 구성하여 검색 시스템을 설계하는데 발행하는 문제들을 해결하였다. Ontology 개념을 도입하여 통합 참여되는 레거시 시스템의 데이터베이스의 메타데이터 스키마와 표준 메타데이터의 1:1매핑뿐만 아니라 1:N매핑관계까지 해결하였고, XMDR Query 템플릿을 이용하여 검색할 때 문제되는 이기종간의 시스템 이질성까지 해결하였다. 그러나 온톨로지 개념의 시소러스 구성과 표준 메타데이터와 매핑시 반자동적으로 매핑 사상 되는 기법을 좀 더 자동적인 매핑 사상이 되도록 구축해야 할 필요성이 있다.

참고문헌

- [1] W.A Ruh et al, Enterprise Application Integration, John Wiley & Sons, New York, 2001
- [2] Joseph M.Firestone, Ph.D., Executive Information Systems, Inc., Portal Progress and Enterprise Content Management, January 9, 2003
- [3] Wang Yan, Law K.C.K., "A Mobile Agent based System for Distributed Database Access on the Internet", IEEE Communication Technology proceedings, Vol.2, pp.1587-1590, Aug., 2000
- [4] McGuiness, D., Fikes, R., Hendler, J. and Stein, L., "DAML+OIL: an ontology language for the Semantic Web", IEEE Intelligent Systems, Vol. 17, No. 5, pp. 72-80, 2002
- [5] David Wang, "Automated Semantic Correlation between Multiple Schema Information Exchange", M.I.T, MM, May, 2000
- [6] <http://xmdr.org>

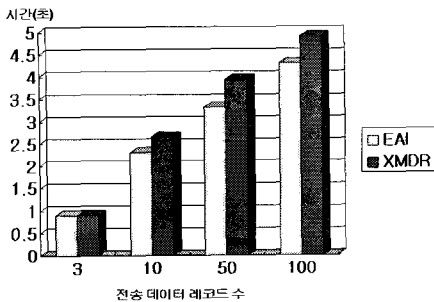


그림 12.1 검색 평균 응답시간 비교분석1

Fig. 12.1 Retrieval average response time comparison1

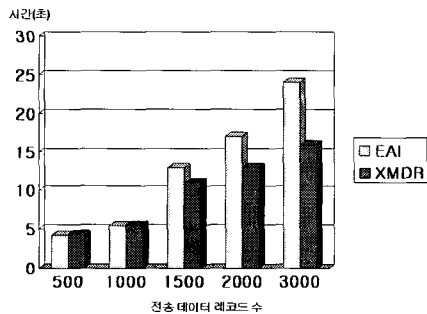


그림 12.2 검색 평균 응답시간 비교분석2

Fig. 12.2 Retrieval average response time comparison2

검색 조건을 파라미터 값을 이용하여 XMDR 서버에 전송하고 매핑과정을 수행하여 변환하는 과정을 거친 후 질의문을 생성하여 실제 데이터베이스의 결과를 전송받게 되므로 처리 시간이 증가한다. 반면 그림12.2와 같이 XMDR과 EAI 환경에서 평균 응답시간을 비교한 결과 검색 평균 응답시간이 EAI 환경보다 다소 작은 것으로 비교 분석 되었다. 이러한 결과를 분석해 보면 EAI 환경의 경우 대량의 검색 데이터인 경우 이기종간의 서버와 클라이언트 간 데이터의 규격을 맞추기 위하여 많은 자원을 소모하게 된다. EAI 내부적인 프로세서와 같이 바이너리 방식 처리와 같이 변환이 필요한 환경에서는 많은 자원이 필요로 하다.

저자소개



문 석 재(Seok-Jae Moon)

2002년 방송통신대학교 전자계산학
졸업

2004년 광운대학교 컴퓨터소프트웨
어학과 석사

2004년~2005년 펠컴정보시스템 주임연구원

2006년~ 현재 광운대학교 컴퓨터과학과 박사과정

※관심분야: XMDR, 데이터 그리드, 상호운용성



정 계 동(Gye-Dong Jung)

1985년 광운대학교 전자계산학 졸업

1992년 광운대학교 산업정보학석사

2000년 광운대학교 컴퓨터과학박사

1993년~2004년 광운대학교 정보과학원 교수

2005년~ 현재 광운대학교 교양학부 교수

※관심분야: XML 분산시스템, 분산 컴퓨팅기술, 이동
에이전트



최 영 근(Young-Keun Choi)

1980년 서울대학교 수학교육과
이학사

1982년 서울대학교 계산통계학과
이학석사

1989년 서울대학교 계산통계학과 이학박사

1983년~ 현재 광운대학교 컴퓨터과학과 교수

1992년~2000년 광운대학교 전산정보원 원장

2002년~2005년 광운대학교 교무연구처장

※관심분야: 객체지향설계, 분산시스템, 이동에이전트,
상호운용성