Journal of The Korea Society of Computer and Information
Vol. 21 No. 10, pp. 107-114, October 2016

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.10.107

Real-time Virtual Integration of heterogeneous system and Union Query System

Seong-Hyeon Min *, Sang-Joon Lee **, Ho-Young Kwak ***

Abstract

In this paper, we propose the building method for the unified IoT service platform between the variety of legacy systems with the heterogeneous structures and the operating environments. To accommodate the structural diversity of legacy systems, we develop a virtual integrated structure and interface abstracting method. By this, the new platform can use the data sources of the legacy systems as if the data source is in the new platform. In a manner without modification of the legacy systems with performing minimum tasks, we proposes a model that can be integrated without affecting the availabilities of the legacy systems.

▸ Keyword : real-time integration, virtual Integration, heterogeneous system, union query, legacy system

I. Introduction

최근 사물인터넷 기술이 빠르게 확산되면서 사물 인터넷 서비스 플랫폼 기술에 대한 관심도 함께 높아지고 있다. 이러한 흐름을 반영하듯 전 세계 주요 표준화기관 및 산업체 컨소시엄들이 사물인터넷 서비스 플랫폼 표준 제정을 통한 주도권 경쟁을 벌이고 있는 중이며, 그 결과로서 oneM2M[1], AllJoyn[2], IoTivity[3]와 같은 사물인터넷 서비스 플랫폼의 구축 및 운영에 관한 표준 규격들이 개발되었고, 현재도 지속적으로 업데이트가 이뤄지고 있다.

이 같은 사물인터넷 서비스 플랫폼에 관한 표준기술들이 주로 다루고 있는 관심사는 앞으로 새롭게 구축할 사물인터넷 생태계에 대한 규격을 제시하는 데 있다. 그러나, 수많은 시스템들이 어느 기구의 표준도 적용하지 않은 채로 구축되어 왔으며, 현재까지도 비표준 기술을 활용하는 시스템들이 많은 실정이라는 점을 고려할 필요가 있다. 데이터 정의에 대한 표준화가 보장되지 않은 상태에서 많은 데이터가 축적되면, 이종 시스템들 간의 정보 공유가 점점 어려워지는 문제가 발생하는데[4], 이는 사물인터넷 서비스 플랫폼들이 활용할 수 있는 데이터 소스의 범위가 좁아진다는 것을 의미한다.

최근 들어, 데이터를 취급하는 대부분의 기술들은 다양한 시스템으로부터 수집된 데이터를 융합 활용하는 방식을 중요하게 고려하고 있다. 예를 들어, 실제 상황에 근접한 상황정보 추론을 위해 다양한 이기종 센서 데이터를 융합하여 활용하는 방식[5,6]이나, 필요한 데이터를 온전히 수집하기 어려운 상황에서의 해결책으로서 데이터 통합(Data fusion) 기법을 활용하는 방식[7] 등이 그것이다.

앞서 언급한 사물인터넷 표준화 기술들 역시 데이터의 공동활용을 중요한 목표로 제시하고 있다는 점을 감안할 때 데이터 소스의 활용범위 확대 능력은 해당 기술의 효용성 평가에 직접적인 영향을 미칠 수도 있다. 그러므로, 사물인터넷 서비스 플랫폼이 데이터 파편화에 따르는 문제를 해소하고 데이터 융합의 효과를 극대화하기 위해서는 과거의 방식으로 구축해 둔 레거시 시스템에 대한 통합 활용 방안도 함께 제시할 수 있어야 할 것이다.

특히, oneM2M 표준과 같이 대형 도메인 및 다중 도메인에

• First Author: Seong-Hyeon Min, Corresponding Author: Sang-Joon Lee
*Seong-Hyeon Min(radiosoop@gmail.com), Smart Convergence R&D Center, SUM Engineering Inc.
**Sang-Joon Lee(sjlee@jejunu.ac.kr), School of Computer Engineering, Jeju National University
***Ho-Young Kwak(kwak@jejunu.ac.kr), School of Computer Engineering, Jeju National University
• Received: 2016. 09. 20, Revised: 2016. 10. 19, Accepted: 2016. 10. 26.
• The reserach was supported by 'Software Convergence Technology Development Program', through the Ministry of Science, ICT and Future Planning(S0170-15-1084)

서의 수평적 서비스 통합을 고려하는 기술들의 경우 레거시 시스템에 대한 통합방안 제시는 필수적이다. oneM2M 표준에서 레거시 시스템 연동을 위해 제시하는 방법을 살펴보면 OMA DA[8], BBF TR-069[9], OMA Lightweight M2M[10]과 같은 기존의 장치관리 기술표준들을 지원하도록 함으로써 레거시 시스템과 새로 구축하는 시스템을 연동할 수 있는 방법을 제공하고 있음을 알 수 있다.

그러나, 대형 시스템의 경우와는 달리 중소형 규모의 레거시 시스템들은 oneM2M 표준에서 지원 대상으로 선정한 기존의 기술표준도 지원하지 않는 경우가 많으며, 흔히 미들웨어 또는 통합서버라는 명칭으로 통용되는 구성요소를 중심으로 독자적인 방식을 통해 데이터 교환 및 장치 관리를 수행하는 구조를 갖고 있는 경우를 실무에서 흔히 접할 수 있다. 또한 레거시 시스템에 포함된 장치들이 최근 사물인터넷 기술 표준들이 제시하는 기능들을 수용하기 어려운 구조나 사양으로 제작되어 있는 경우도 많다.

본 연구에서는 각기 다른 구조와 운영환경을 갖고 있는 다양한 레거시 시스템들을 제한적으로나마 신규 시스템과 연동하여 통합된 사물인터넷 서비스 플랫폼으로 구축할 수 있게 하는 방법에 관해 다룬다. 이 때, 통합 대상 레거시 시스템은 RDBMS를 이용하여 데이터를 관리하고 있는 경우를 가정하여 연구를 진행하며, 결과적으로는 다수의 통합 대상 레거시 시스템들의 RDBMS 내에 저장된 데이터들을 해당 시스템의 구조적 특성과 무관하게 통합하여 활용할 수 있는 모델을 제시한다.

이를 위해, 레거시 시스템들이 갖는 구조적 다양성을 수용할 수 있도록 데이터 조회 인터페이스를 추상화함으로써, 레거시 시스템의 데이터 소스가 신규 플랫폼 내에 존재하는 것처럼 가상 통합을 이루는 구조를 개발한다. 이 과정에서 CUAHSI ODM[11]이 제시하는 데이터 모델 설계 콘셉트를 참조한다. 이는 모든 유형의 관측 데이터를 관측일시, 관측장소, 관측유형이라는 속성이 결합된 형태로 취급함으로써 형식적 통일을 추구하는 방법이다.

또한, 레거시 시스템들이 저마다 다른 운영체제와 DBMS 종류, DB 스키마를 갖고 있음에도 범용 프로그램을 이용하여 간단한 설정만으로 데이터를 통합하는 방법을 구현하기 위하여 이종 데이터 소스에 대한 통합 솔루션인 JBoss Data Virtualization[12]의 설계 콘셉트 또한 참조한다.

이렇게 설계된 모델을 활용하면, 레거시 시스템에 대한 수정 없이 최소한의 작업만으로 연동을 수행하여, 레거시 시스템의 운영에 영향을 주지 않으면서도 전체적으로 활용할 수 있는 데이터의 종류 및 규모를 확장시킬 수 있게 된다. 이는 기존 플랫폼 구조에 비해 데이터 접근성이 향상된 형태로서 더 다양한 서비스 개발이 가능한 플랫폼을 구축할 수 있게 한다.

2장에서는 oneM2M 표준이 제시하는 기존시스템 연동방안과 그 외 데이터 통합 관련 기술들을 연구한다. 3장에서는 통합 데이터 모델의 콘셉트와 구현내용을, 4장에서는 실험 결과를 각각 설명한다. 5장에서는 결론 및 향후 연구 주제를 제시한다.

II. Related works

1. oneM2M Standards

사물인터넷 생태계에서 수평적 서비스 융합이 가능한 플랫폼의 구축 및 운영에 관한 기술표준인 oneM2M[1]은 M2M 장치 관리 기술을 직접 개발하는 대신, OMA DA[8], BBF TR-069[9], OMA Lightweight M2M[10]과 같이 이미 상용화된 완성도 높은 기존 기술들을 선정하여 플랫폼이 이를 지원할 수 있게 하는 전략을 제시하였다[13]. Fig. 1은 oneM2M의 서비스 레이어별로 연동 가능한 기존 장치 관리 기술들을 나타낸다.

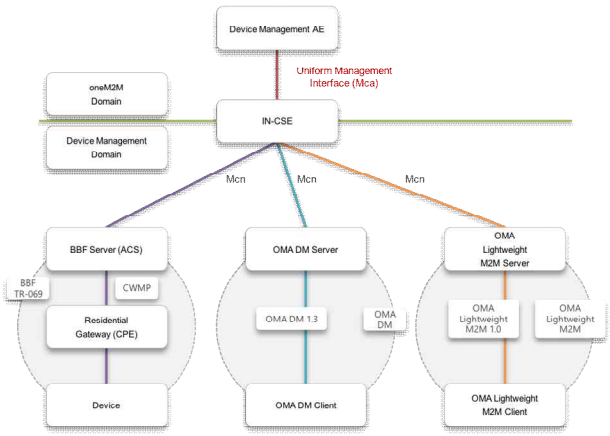

Fig. 1. Legacy device management technology and oneM2M service layer[13]

그러나, 실제 업무 환경에서의 레거시 시스템들은 그러한 기존의 표준기술들조차 적용되지 않았거나, 적용할 수 없는 경우들이 존재한다. oneM2M은 그러한 시스템들의 데이터를 활용할 수 있는 방법을 제공하지 않는다. 본 연구에서는 이런 경우 장치를 직접 관리할 수 없더라도 해당 시스템이 관리하는 DBMS와의 연동을 통해 제한적으로나마 해당 데이터소스를 활용할 수 있는 방안을 제시한다. 이는 본 연구의 핵심 아이디어이며, 이를 위해 이종 데이터 간 실시간 가상 통합 모델을 제시한다.

2. Red Hat JBoss Data Virtualization

JBoss Data Virtualization[12]은 Red Hat사가 개발한 데이터 공급 및 통합 솔루션으로서, 흩어져 있는 이종의 데이터 소스를 단일 소스로 처리할 수 있도록 가상 통합 기능을 제공한다. 이 솔루션은 Fig. 2와 같이 RDBMS, NoSQL, File, SaaS, Hadoop 등 다양한 형태로 존재하는 데이터 소스에 대한 유연한 통합을 가능하게 하며, 이렇게 통합된 데이터 소스를 다시 일관성 있는 오픈 표준 인터페이스로 전환할 수 있게 한다. 이 솔루션의 도입을 위해서는 사용자가 데이터 통합을 위해 실제로 사용하고자 하는 Data View의 스키마를 정의하고 레거시 데이터 소스와의 매핑 규칙을 개발하는 하는 과정이 필요한데,

이 과정에서 분석 및 설계에 대한 노력이 요구된다.

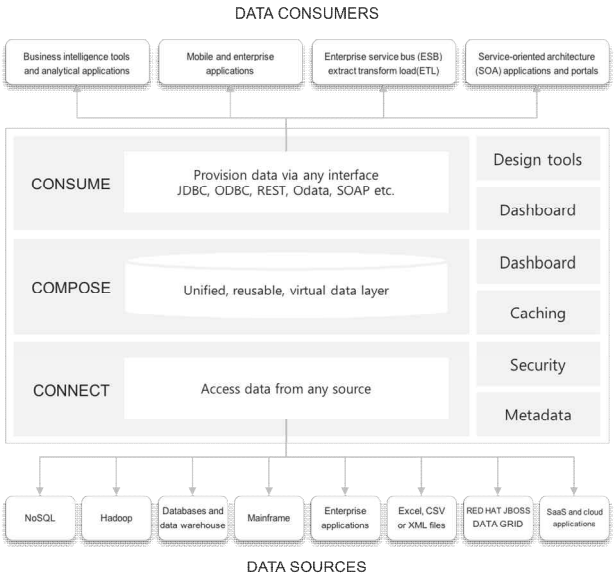

Fig. 2. JBoss Data Virtualization

본 연구에서는 그러한 과정을 생략하거나 최소화할 수 있도록 사물인터넷 환경에서 범용으로 사용할 수 있는 데이터모델을 사전 정의하여 데이터 통합 시 활용할 수 있게 한다. 따라서, JBoss Data Virtualization이 다양한 유형의 레거시 데이터 소스에 접근할 때 사용하는 방식은 참고하되, 사물인터넷 환경에서 취급하는 Thing, Sensor 등의 개념과 관측데이터 취급에 필요한 개념들을 레거시 시스템의 DB Entity들에 사상할 수 있도록 사전 정의된 데이터 모델과의 매핑 규칙을 제공함으로써, 관측데이터를 주로 활용하는 사물인터넷 서비스 플랫폼 구축 시 보다 간편하게 통합을 구현할 수 있도록 하는 데 연구의 초점을 맞춘다.

3. CUAHSI ODM

수문학(hydrology) 연구단체인 CUAHSI(The Consortium of Universities for the Advancement of Hydrologic Science, Inc.)[14]는 수문 연구용으로 수집되는 수많은 이종 데이터를 효과적으로 활용할 수 있도록 ODM(Observations Data Model)[11]이라는 수문 연구용 통합 데이터 모델을 개발했다.

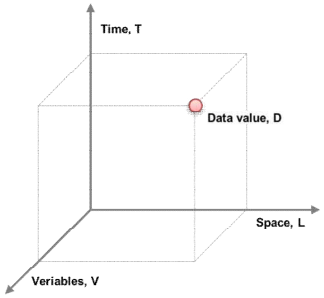

Fig. 3. Data identification of CUAHSI ODM

이 모델은 모든 유형의 관측 데이터를 '시간값', '공간값', '변수값(관측유형)'이라는 세 가지 요소의 결합체로 취급하며, 이종의 데이터들이라도 그 실체는 같은 구조로 정의되므로 DBMS상에서 DataValues라는 하나의 테이블 구조를 중심으로 일관성 있게 관리될 수 있다. Fig. 3은 데이터에 대한 ODM의 관점을 표현하고 있다.

측정된 값 외의 다른 부가 정보들도 Fig. 4에서 보이는 것과 같이 연관된 테이블에 저장되므로 사용자는 필요에 따라 원하는 조건을 만족하는 값들을 용이하게 조회할 수 있으며, 관측장치나 시스템 종류와 상관없이 특정 시간과 공간 범위를 조건으로 하여 모든 값들을 가져와서 분석할 수 있다는 점이 가장 큰 효용이라고 할 수 있다.

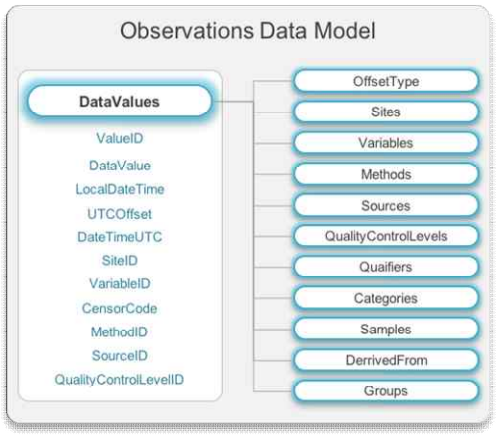

Fig. 4. CUAHSI ODM Overview

본 연구에서는 이러한 데이터 모델 설계 콘셉트를 활용하고, 여기에 레거시 시스템의 데이터 스키마를 사상하여 활용하는 방식으로 가상 통합 모델을 설계한다.

III. Real-time Virtual Integration Model

1. Real-time Virtual Integration Concept

본 연구에서 제시한 이종 시스템의 실시간 가상 통합 및 통합 쿼리 시스템의 설계 콘셉트는 Fig. 5와 같다. 이 시스템은 하나의 Integration Server(이하 통합서버)와 다수의 Data Integration Agent(이하 DIA)의 연결로 구성된다.

통합서버는 여러 시스템에 흩어져 있는 데이터에 대한 통합관리를 가능하게 하며, 통합서버의 핵심 구성요소는 통합관리모듈과 통합쿼리모듈이다. 통합관리모듈은 통합대상들의 정보를 관리하며, 통합쿼리모듈은 쿼리를 실제로 수행할 레거시 시스템들을 찾아서 사용자로부터 입력받은 쿼리조건을 전달한다.

DIA는 통합 대상 레거시 시스템의 DB서버에 설치되는 연동 프로그램으로서, 통합서버로부터 해당 레거시 시스템의 정보를 전달받아 동적으로 쿼리를 생성하고 레거시 데이터를 표준 데

이터 모델용으로 변환 후 반환하는 기능을 수행한다.

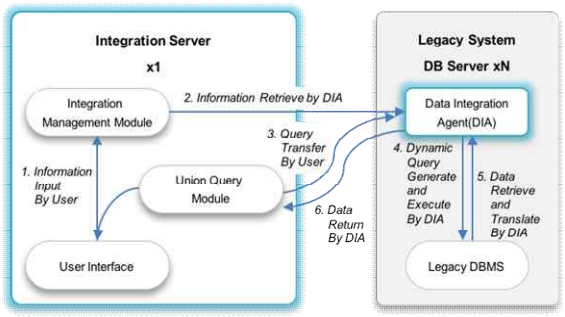

Fig. 5. Real-time Virtual Integration Concept

이 시스템에서의 데이터 흐름은 레거시 시스템에 대한 정보 입력으로부터 시작된다. 이렇게 입력된 정보는 DIA에 전달되어 연동설정이 구성된다. 이 상태에서 사용자가 데이터 조회 명령을 입력하면 이 명령이 통합 쿼리 모듈에 의해 해당 데이터를 보유한 시스템의 DIA로 전달되고, DIA는 해당 시스템의 환경 정보를 가지고 동적으로 쿼리를 구성하여 실행한 후 조회된 데이터를 통합 쿼리 모듈을 통해 사용자에게 반환한다.

Table 1. Union Query Example

Query Type	Query Statement
Zipcode based	select measurementType in (waterlevel,rainfall) and measurementDatetime in (20150101-20150228) and zipcode in (61253-61255)
GPS based	select measurementType in (waterlevel,rainfall) and measurementDatetime in (20150101-20150228) and position in (34.125,126.532,10)

이 때, 사용자는 찾고자 하는 데이터가 어떤 시스템에 어떤 형태로 존재하고 있는지 모르더라도 관측유형, 관측일시, 관측장소를 조건으로 하여 쿼리를 작성할 수 있는데, 이처럼 실제로는 분리되어 있는 시스템들을 논리적으로 통합된 것처럼 활용할 수 있게 하는 방법을 본 연구에서는 '실시간 가상 통합'이라 하며, Table 1의 예시와 같이 데이터를 보유한 레거시 시스템의 종류나 데이터베이스의 구조에 대해서는 신경 쓰지 않고도 필요한 조건만으로 질의하게 하는 방법을 '통합 쿼리'라 한다.

이러한 통합 쿼리를 가능하게 하는 구성요소인 통합서버와 DIA의 구현 및 실험결과에 대해서는 이어지는 항에서 상세하게 설명한다.

2. Implementation of Integration Server

통합서버는 사용자 인터페이스, 통합 관리 모듈, 통합 쿼리 모듈을 포함하는 웹 기반 애플리케이션으로 구현하였으며, 통합서버 개발 및 실행환경은 Table 2와 같다.

통합서버는 레거시 시스템 연동에 필요한 정보들을 관리하는 기능을 필요로 하는데, 이러한 기능들은 '통합 관리 모듈'로서 구현하였다.

Table 2. Development and Experiment Environment

Div.	Information	Description
Development Environment	Programing Language	JAVA (JDK 7)
	Web Dev. Framework	Spring Framework 3
	JPA Provider	Hibernate 4
Execution Environment	Operating System	Cent OS 7
	Servlet Container	Tomcat 7
	DBMS	MySQL 5

이 때, 연동 설정을 위해 관리해야 하는 주요 정보 항목은 Table 3과 같다. Table 3에 나타난 정보들은 레거시 시스템에 접근하기 위해 필요한 정보들과, 레거시 데이터로부터 조회된 값을 표준형식으로 사상하기 위해 필요한 정보들의 조합이다. 이 정보들을 통해서 형식이 다른 데이터들을 일관된 형태로 변환하여 활용할 수 있게 된다.

Table 3. Legacy System Information for Integration

Div.	Information	Description
System Info.	macAddress	macAddress of legacy system
	sensors	information of sensors in legacy DB(ID, measurement position, measurement type)
	dialect	DBMS Type(MySQL, Oracle, etc.)
DB Scheme	tableName	target table's name
	valueFieldName	value field's name
	datetimeFieldName	datetime field's name
	datetimeFormatPattern	datetime format pattern ex:yy-MM-dd HH:mm:ss
	variableInfo	measurement type ex: temperature, humidity, etc.
	positionInfo	measurement position ex: GPS position, zipcode, etc.
	unitTransformCoefficient	a multiply Coefficient that will needed value standardization ex: cm → m : x0.01, kg → g : x0.001
DB Access Info.	databaseUrl	DB access path
	databaseUsername	username for DB access
	databasePassword	password for DB access

이 정보들을 입력하고 나면 레거시 시스템의 DB 서버에 DIA를 설치할 수 있다. 이때, 레거시 시스템에서의 작업 요소를 최소화 하기 위해 설정과정도 외부에서 진행될 수 있게 구성하였는데 '통합 관리 모듈'이 관리하는 Table 3의 정보들이 DIA에도 전달된다. DIA는 매 실행시점마다 통합서버에 접근하여 자신이 접근해야 할 DB에 대한 접속정보 및 DB 스키마 정보를 읽어들인 후 데이터 접근에 필요한 준비 작업을 수행하는 방식이다. 통합서버는 정보를 요청하는 DIA의 macAddress를 이용하여 해당정보를 찾아 전달하도록 구현하였다.

이렇게 레거시 시스템에 대한 정보가 사용자에 의해 입력되고, 해당 레거시 시스템의 DB 서버에 DIA가 설치되면 실시간 가상 통합이 완료된 상태가 된다.

통합서버의 또 다른 구성요소로 '통합 쿼리 모듈'이 있다. 이 모듈은 사용자가 데이터 조회 명령을 입력했을 때 레거시 시스템들을 가상으로 통합하여 일괄적으로 조회를 수행하도록 구현하였다. '통합 쿼리 모듈'은 '통합 관리 모듈'에 의해 관리되고 있는 레거시 시스템들의 정보를 확인하여 조건에 부합하는 시스템 목록을 작성하는 기능과 해당 시스템들을 대상으로 원격 쿼리를 실행하고, 결과를 취합하여 사용자에게 반환하는 기능을 차례대로 수행한다.

이와 같이 전술한 구성요소들의 기능을 활용하면 사용자는 원하는 데이터의 속성만을 정의하는 방식으로 통합 쿼리를 실행할 수 있다. 여기서 데이터의 속성이란 관측유형, 관측시점, 관측위치를 의미한다. 이 시스템은 '통합 관리 모듈'과 '통합 쿼리 모듈'을 통해 데이터에 대한 추상화를 제공하며, 레거시 시스템의 데이터들은 이 추상화 모델에 맞게 사상되도록 설계되어 있으므로, 사용자는 원하는 데이터가 어느 시스템에 어떤 구조로 저장되어 있는지를 신경 쓰지 않고도 해당 속성을 지정하는 것만으로 필요 데이터의 조합을 쿼리 결과로 얻을 수 있다.

3. Implementation of DIA

통합서버와 레거시 시스템의 연동을 위해 레거시 시스템 내에 설치하는 프로그램인 DIA를 추가로 구현하였으며, Fig. 6는 DIA의 내부 구성요소들을 표현한다. DIA는 레거시 DBMS에 직접 접근하는 프로그램으로서, DIA를 구현함에 있어서 가장 중요한 고려 요건은 레거시 시스템의 운영환경에 대한 수정은 최소화하면서 설치 및 실행이 가능해야 한다는 점이었다. 또한, 레거시 시스템들의 운영환경이 모두 상이하므로 DIA의 소스코드를 수정하지 않고 쉽게 적용할 수 있도록 범용성을 강화하는 방안도 고려하였다. 이를 위해 DIA는 자바 언어를 이용하여 작성하였다. 이는 자바 실행환경이 구동될 수 있는 시스템이라면 무리 없이 설치와 실행이 가능하다는 것을 의미한다. 그리고, 별도의 설치 과정 없이 단순하게 프로그램 파일을 해당 시스템 내에 복사해서 바로 실행할 수 있는 구조로 구현하였다.

이와 같은 맥락에서, 레거시 시스템의 구조에 맞게 DIA 자체의 구동환경을 설정하는 일도 레거시 시스템 내에서 수행되지 않도록 통합서버에 저장된 정보를 DIA의 Connection Configurator가 읽어오는 방식으로 구현하였다.

DIA는 통합서버의 '통합 쿼리 모듈'로부터 수신되는 쿼리 요청을 처리할 수 있어야 하는데, 이를 위해 RESTful 방식의 API 제공이 가능한 Web 기반 애플리케이션으로 구현하였다. 다만, 이 경우 Servlet Container 설치 과정을 생략하고, 성능부하를 최소화하기 위해 경량 Servlet Container인 Embedded Tomcat을 프로그램에 내장하는 방식을 적용했다. 이는 프로그램이 실행될 때, 내장 Servlet Container도 함께 실행되는 방식이다.

웹 API를 통해 수신된 쿼리 요청은 Java Persistence API[15] Provider인 Hibernate[16]를 통해 레거시 DBMS에 맞는 쿼리문으로 자동 변환 및 실행되도록 구현했다. Hibernate는 현존하는 대부분의 주요 RDBMS에 대한 맞춤형 쿼리문 생성을 지원할 수 있다. Hibernate의 이러한 기능을 이용하여 Connection Configurator가 '통합 서버'로부터 읽어들인 레거시 DBMS의 유형 및 스키마 정보를 이용하여 레거시 데이터를 본 시스템에서 미리 정의해 놓은 데이터 스키마에 맞게 변환하여 결과로 반환하는 기능을 구현했다. 이 때, 데이터 변환은 시간 형식 통일과 값 단위(예: kg → g, mm → m 등) 통일 작업을 포함하여, 사용자에 의한 데이터 정제 과정을 최소화할 수 있게 구현하였다.

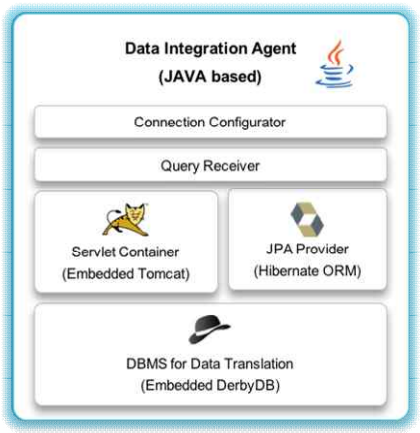

Fig. 6. Structure of Data Integration Agent

또한, 외부 요청에 빠르게 응답할 수 있도록 필요시 레거시 데이터를 미리 변환하여 내장DB인 DerbyDB[17]에 저장할 수 있도록 구현했다. DerbyDB는 Embedded Tomcat과 마찬가지로 프로그램에 내장할 수 있는 경량 오픈소스 DBMS이다.

IV. Experiments

본 연구에서 제안하고 구현한 시스템에 대한 실험을 두 가지 주제로 구분하여 수행하였다. 먼저, 레거시 시스템 연동에 필요한 프로그램인 DIA가 설정 변경에 적절하게 반응하는 지를 실험하고, 이어서 통합 쿼리를 수행하여 정확한 결과를 반환하는지 실험하였다.

1. Data Translation of DIA

DIA가 설계 시 의도된 대로 동작하는 지 실험하였다. 실험은 DIA가 자동으로 통합서버로부터 설정 정보를 읽어들이고, 레거시 시스템들의 다양한 데이터 저장환경에 동적으로 적응하며, 정확하게 데이터를 조회하여 표준 형식으로 변환하는 동작을 수행하는지 확인하는 과정으로 구성하였다. 실험 개요는 Table 4와 같다.

이 실험의 핵심은 DIA가 소스코드나 설정파일 수정 없이 통합서버에서 제공하는 다양한 설정정보들을 정확하게 반영할 수 있는 지 확인하는 것이므로, 하나의 시스템에 DIA를 설치해두고 DIA가 참조하는 통합서버의 레거시 시스템 설정정보를 순차적으로 변경하였다. 이 때 변경되는 정보는 각기 다른 종류의 DBMS와 스키마로 조합된 3종의 설정 세트이다.

Table 4. Data Translation Testing overview

job	description
prepare three experimental legacy databases	scheme :table name, fields name, datetime etc. DBMS : MySQL 2sets, PostgresQL 1set
enter the sensor information to collect on integration server	legacy system's access information (DB path, username, password etc.) legacy system's sensor information (sensor ID, type, position etc.)
check the data in embedded DB while chainging the sensor informaion	experimental bus information system data transformed result experimental parking information system data transformed result experimental water information system data transformed result

Fig. 7에서 보듯이 DIA가 변경되는 설정정보를 적시에 읽어들여 사전 정의된 표준 형식에 맞게 데이터를 변환 후 내장DB에 정상적으로 저장하는 것을 확인하였다. 이는 실시간 가상 통합이 완료되었음을 의미하며, 이렇게 저장된 데이터를 통해 사용자의 통합 쿼리 명령에 신속하게 데이터를 반환할 수 있는 상태가 되었음을 확인할 수 있었다.

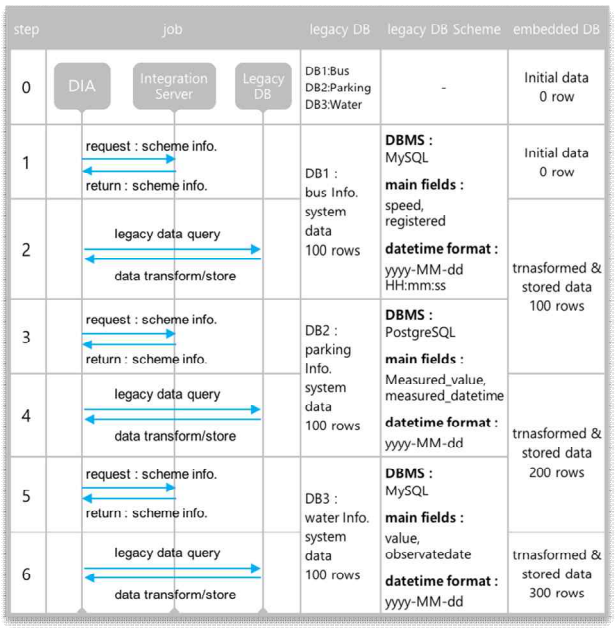

Fig. 7. Flow of Data Translation Test

2. Union Query

실험을 통해 DIA가 정확하게 동작하고, 시스템 가상 통합이 완료되었음을 확인하였다. 이에 사용자 관점에서 통합 쿼리를 수행하는 과정에 대한 실험을 추가로 진행하였다.

Table 5. Legacy systems for Real-time Union Query

Legacy Systems	Description
System #1 (sensorId : 1)	• OS : Windows Server 2008 • DBMS : MySQL 5 • Data measurementType : Waterlevel measurementDatetime : 20150630~20151231 measurementZipcode : 50000
System #2 (sensorId : 2)	• OS : Cent OS 7 • DBMS : MySQL 5 • Data measurementType : Rainfall measurementDatetime : 20150930~20160331 measurementZipcode : 50001
System #3 (sensorId : 3)	• OS : Mac OS X 10.11 • DBMS : PostgreSQL 9.4 • Data measurementType : Rainfall measurementDatetime : 20160101~20160630 measurementZipcode : 50002

Table 5와 같이 실시간 통합 쿼리 대상이 되는 레거시 시스템들을 실험용으로 구성하고 통합서버를 이용하여 조건을 변경해가면서 데이터 조회를 시도하였다. 용이한 결과 확인을 위해 레거시 시스템 1대당 보유센서 1개씩의 센서 데이터가 통합서버에 연결되도록 구성하였고, 센서 식별자는 레거시 시스템 번호와 같게 부여하였다.

Fig. 8은 조건별 데이터 조회 기능을 제공하는 통합 쿼리용 사용자 인터페이스 구현 결과이며, 이를 이용하여 실험을 진행했다.

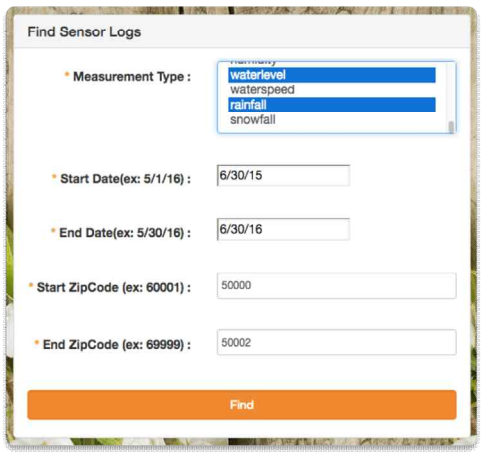

Fig. 8. User Interface for Union Query

통합쿼리 실험에 사용한 조건들과 실제 수행결과를 Table 6에 정리하였다. 여기에서 제시된 조건과 결과들은 다음과 같은 의미를 갖는다. Test 1은 아무 제약 없이 모든 레거시 시스템

으로부터 데이터를 가져오는지를 확인하기 위한 것이다. 이를 위해 모든 실험용 레거시 시스템의 데이터가 해당되는 조건을 사용했고, 결과적으로 시스템은 모든 레거시 시스템의 데이터를 취합하여 반환했다. 결과로 제시된 데이터 목록에서 확인해야 하는 컬럼은 'sensorId'와 'stationId'이다. 'stationId'는 레거시 시스템에서 직접 관리하는 센서들의 식별자이며, 'sensorId'는 레거시 서버의 종류와 상관 없이 통합서버에서 새롭게 부여한 식별자이다. 따라서, 조건 변경에 따라 어느 레거시 시스템에서 데이터를 가져오는 지 확인하기 위해서 각 결과에 나타난 'sensorId'를 확인하였다.

Test 2는 관측유형을 제약으로 사용한 경우이다. 모든 시스템에서 강우량 관측 결과를 모두 가져오는 조건을 설정했고, 그 결과로 #2 및 #3 시스템에서 데이터가 반환되었다. 마찬가지로 Test 3, 4에서는 각각 관측일시 및 관측위치를 제약으로 사용하였고 조건에 부합하는 데이터가 해당 시스템으로부터 반환되었음을 확인하였다.

Table 6. Result of Real-time Union Query

<table>
<tr><td rowspan="2">Test 1</td><td>Query Condition : Type in (waterlevel,rainfall)
Datetime in (20150630-20160630)
Zipcode in (50000-50002)</td></tr>
<tr><td>result data from system #1,#2,#3
<table><tr><th>measuredDatetime</th><th>measuredValue</th><th>sensorId</th><th>stationId</th></tr><tr><td>2015-06-30 00:00:00</td><td>0.231</td><td>1</td><td>7</td></tr><tr><td>2015-06-30 00:00:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:00:00</td><td>6.5</td><td>3</td><td>53</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.226</td><td>1</td><td>7</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:10:00</td><td>3.5</td><td>3</td><td>53</td></tr></table></td></tr>
<tr><td rowspan="2">Test 2</td><td>Query Condition : Type in (rainfall)
Datetime in (20150630-20160630)
Zipcode in (50000-50002)</td></tr>
<tr><td>result data from system #2,#3
<table><tr><th>measuredDatetime</th><th>measuredValue</th><th>sensorId</th><th>stationId</th></tr><tr><td>2015-06-30 00:00:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:00:00</td><td>6.5</td><td>3</td><td>53</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:10:00</td><td>3.5</td><td>3</td><td>53</td></tr></table></td></tr>
<tr><td rowspan="2">Test 3</td><td>Query Condition : Type in (waterlevel,rainfall)
Datetime in (20150630-20151231)
Zipcode in (50000-50002)</td></tr>
<tr><td>result data from system #1,#2
<table><tr><th>measuredDatetime</th><th>measuredValue</th><th>sensorId</th><th>stationId</th></tr><tr><td>2015-06-30 00:00:00</td><td>0.231</td><td>1</td><td>7</td></tr><tr><td>2015-06-30 00:00:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.226</td><td>1</td><td>7</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.5</td><td>2</td><td>184</td></tr></table></td></tr>
<tr><td rowspan="2">Test 4</td><td>Query Condition : Type in (waterlevel,rainfall)
Datetime in (20150630-20151231)
Zipcode in (50001-50002)</td></tr>
<tr><td>result data from system #2,#3
<table><tr><th>measuredDatetime</th><th>measuredValue</th><th>sensorId</th><th>stationId</th></tr><tr><td>2015-06-30 00:00:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:00:00</td><td>6.5</td><td>3</td><td>53</td></tr><tr><td>2015-06-30 00:10:00</td><td>0.5</td><td>2</td><td>184</td></tr><tr><td>2015-06-30 00:10:00</td><td>3.5</td><td>3</td><td>53</td></tr></table></td></tr>
</table>

3. Means of Experiments

이러한 실험을 통해 확인할 수 있는 기존 연구와의 차별성은 다음과 같다. 먼저, oneM2M 표준[1]이 레거시 시스템 연동을 위해 기존 장치관리 표준기술들에 대한 인터페이스를 제공하는 것과 달리, 본 연구에서 제안하는 모델은 레거시 시스템이 비표준 기술을 사용하더라도, 그 데이터를 활용할 수 있는 방법을 제공할 수 있다.

뿐만 아니라, 제안하는 모델은 CUAHSI ODM[11]의 관측 데이터 모델링 아이디어를 바탕으로 사물인터넷 환경에 적합한 데이터 관리 형식을 사전에 정의하여 제공하므로, JBoss Data Virtualization[12]을 이용할 때 필요한 과정인 '목표 데이터세트의 형식 정의 작업' 없이도 정상적으로 데이터 통합과 활용이 가능하다.

V. Conclusions

본 연구에서는 다양한 운영체제, DBMS 종류, DB 스키마 구조를 갖는 서로 다른 레거시 시스템들을 논리적으로 가상 통합하여 통합 쿼리를 수행할 수 있는 모델을 제시하였다. 또한, 제안한 모델의 기능을 검증하기 위해서 통합서버 및 DIA를 구현하고 실험하였다.

실험은 두 가지 주제로 구분하여 진행했으며, DIA에 대해 실험한 결과 프로그램 자체 수정이나 설치 환경 변경 없이 레거시 시스템에 간편하게 설치하여 실행하는 것만으로 이종 시스템의 실시간 가상 통합이 가능함을 확인하였다.

그리고, 실험용 레거시 시스템들과 통합서버를 이용하여 조건을 변경해가며 통합 쿼리 기능을 실험하였다. 이 실험을 통해 물리적으로 분리되어 있는 이종 시스템의 이종 데이터들을 가상으로 통합하여 관측일시, 관측위치, 관측유형 요소 지정만으로 데이터를 통합 조회할 수 있음을 확인하였다.

이와 같이 본 연구를 통해 사물인터넷 플랫폼 구축 시 레거시 시스템을 가상으로 통합하고, 이를 기반으로 통합 쿼리를 수행할 수 있게 하는 모델의 제시와 검증을 완료하였다. 이러한 모델을 활용하면 보다 용이하게 레거시 시스템의 데이터를 활용할 수 있으며, 이를 통한 데이터 융합 범위 확대 및 응용 서비스 고도화를 기대할 수 있다.

현재는 RDBMS에 저장된 데이터를 통합하는 기능만을 고려하였으나, 향후, DIA의 기능 확장을 통해 NoSQL, File, Hadoop Storage 형식의 데이터를 연동하는 기능을 제공하는 방법도 연구할 필요가 있다. 또한, oneM2M 표준과 같은 사물인터넷 서비스 플랫폼 표준 기술에서 제시하는 아키텍처에 연동할 수 있는 인터페이스 개발에 관해서도 추가적인 연구가 필요하다.

REFERENCES

[1] oneM2M, *http://www.onem2m.org/*

[2] AllJoyn, *https://allseenalliance.org/framework*

[3] IoTivity, *https://www.iotivity.org/*

[4] Bong-Ki Ahn, "A Data Standardization Method for DB Integration," Master's Thesis, Graduate School of Information Sciences, Soongsil University, 2010

[5] Dong-Hyok Suh, Chang-Keun Ryu, "Multi-sensor Data Fusion Using Weighting Method based on Event Frequency," The Korea Institute of Electronic Communication Sciences, Vol. 6, No. 4, pp. 581-587, August 2011

[6] Eui-Hyuk Lee, "Short Range Target Tracking Based on Data Fusion Method Using Asynchronous Dissimilar Sensors," IEIE(Journal of the Institute of Electronics and Information Engineers), Vol. 49, No. 9, pp. 335-343, September 2012

[7] Jin-Uk Hong, "A study On Data Fusion Using Statistical Matching," Master's Thesis, Graduate School, Sungkyunkwan University, 2014

[8] OMA DM, *http://openmobilealliance.org/about-oma/work-program/device-management/*

[9] BBF TR-069, *https://www.broadband-forum.org/cwmp.php*

[10] OMA Lightweight M2M, *http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0*

[11] David G. Tarboton, Jeffery S. Horsburgh, David R. Maidment, "CUAHSI Community Observations Data Model(ODM) Version 1.1 Design Specifications," CUAHSI, May 2008

[12] Red Hat JBoss Data Virtualization, *https://www.redhat.com/ko/technologies/jboss-middleware/data-virtualization*

[13] J.S. Song, G.M. Lee, J.U. Seo, N.H. Gang, J.H. Kim, S.C. Choi, H.S. Yang, H.D. Choi, S.M. Jeong, S.Y. Kim, H.B. Ahn, "oneM2M service platform standard commentary," Telecommunications Technology Association, 121-132, November 2014

[14] Consortium of Universities for the Advancement of Hydrologic Science, Inc., https://www.cuahsi.org/

[15] Java Persistence API, *https://en.wikipedia.org/wiki/Java_Persistence_API*

[16] Hibernate ORM, http://hibernate.org/orm/

[17] Apache Derby, https://db.apache.org/derby/

Authors

Seong-Hyeon Min received the M.S. degree in Computer Engineering from Jeju National University, Korea, in 2013. He is working at Smart Convergence R&D Center, SUM Engineering since 2009. His interests in IoT Service Platform, intelligent system.

Sang-Joon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Chung-Ang University, Korea, in 1984, 1989 and 1992, respectively. Dr. Lee joined the faculty of the Department of Computer Engineering at Jeju National University, Jeju, Korea, in 1992. He is currently a Professor in the Department of Computer Engineering, Jeju National University. He is interested in intelligent system, computer algorithm.

Ho-Young Kwak received the B.S., M.S. and Ph.D. degrees in Computer Science from Hong-Ik University, Korea, in 1983, 1985 and 1990, respectively. Dr. Kwak joined the faculty of the Department of Computer Engineering at Jeju National University, Jeju, Korea, in 1990. He is currently a Professor in the Department of Computer Engineering, Jeju National University. He is interested in IT-Medical convergence, USN, software system.