



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

博士學位論文

데이터 활용성 강화를 위한  
사물인터넷 서비스 플랫폼 확장

濟州大學校 大學院

컴퓨터工學科

閔 盛 炫

2017 年 2 月

# 데이터 활용성 강화를 위한 사물인터넷 서비스 플랫폼 확장

指導教授 李 尙 俊

閔 盛 炫

이 論文을 컴퓨터工學 博士學位 論文으로 提出함

2016 年 12 月

閔盛炫의 工學 博士學位 論文을 認准함

審査委員長

郭 鎬 榮

委 員

이 상 준 이 시 영

委 員

김 드 현 민

委 員

김 한 일 김 한 인

委 員

박 종 희 박 영 화

濟州大學校 大學院

2016 年 12 月



# An Expansion of IoT Service Platform to Enhance Data Usability

Seong-Hyeon Min  
(Supervised by professor Sang-Joon Lee)

A thesis submitted in partial fulfillment of the requirement for the  
degree of Doctor of Computer Engineering

2016. 12.

This thesis has been examined and approved.

Thesis director, *Hoyoung Kwak*  
Thesis director, *Sang Joon Lee*  
Thesis director, *Dohyeun Kim*  
Thesis director, *Hanil Kim*  
Thesis director, *Chung Hee Park*

December 2016

Department of Computer Engineering  
GRADUATE SCHOOL  
JEJU NATIONAL UNIVERSITY





# 목 차

목 차	i
그림목차	iv
표 목 차	vi
국문초록	vii
Abstract	viii
Abbreviations	x
<b>I. 서 론</b>	<b>1</b>
1. 연구 배경 및 목적	1
2. 논문의 구성	7
<b>II. 관련 연구</b>	<b>8</b>
1. 사물인터넷 데이터 검색	8
1) oneM2M 표준의 자원 발견 방법	8
2) Mobius Platform의 Global Discovery Platform	13
3) oneM2M 표준 자원 접근방식 확장	15
4) 육하원칙 기반 상황인지 데이터 모델	17
2. 이종 시스템 데이터 통합	20
1) 연합 DB 시스템	20
2) CUAHSI 관측데이터 통합 모델	23
3) oneM2M 표준의 레거시 시스템 연동	25
4) 가상화 기반의 데이터 통합	27
5) 데이터 통합을 위한 전역 스키마 생성 기법	29

### III. 서비스 플랫폼 확장 설계 31

1. 제안된 확장 기능 .....	31
2. 육하원칙 기반 IoT 데이터 검색 .....	35
1) 검색 기준 확장 요소 도출 .....	35
2) 데이터 관리 모델 .....	37
3) 육하원칙 기반의 데이터 검색 기법 .....	39
3. 에이전트 기반 IoT 연합 DB 구성 .....	42
1) 제안된 연합 DB 구성 방안 .....	43
2) 데이터 연합 관리 시스템 .....	43
3) 데이터 연합 에이전트 .....	44
4) IoT 연합 DB 기반 통합 질의 .....	44
4. IoT 데이터의 통계적 처리 .....	46
1) 제안된 통계 처리용 데이터 모델 .....	46
2) 시간대별 기초통계 자동 사상 .....	47
3) 기초통계 활용 프로세스 .....	48

### IV. 확장 기능 구현 50

1. 기능 구현 환경 .....	50
2. 육하원칙 기반 IoT 데이터 검색 기능 .....	54
3. 에이전트 기반 IoT 연합 DB 구성 기능 .....	58
1) 데이터 연합 관리 시스템 .....	58
2) 데이터 연합 에이전트 .....	62
4. IoT 데이터의 통계적 처리 기능 .....	65

<b>V. 실험 및 평가</b>	<b>69</b>
1. DFA의 데이터 변환 .....	69
2. 통합 질의 .....	72
3. 자동 기초 분석 .....	76
4. 실험 결과 평가 .....	81
<b>VI. 결 론</b>	<b>85</b>
<b>참고문헌</b>	<b>87</b>

# 그림 목 차

Fig. 1. Horizontal platform vision of oneM2M .....	2
Fig. 2. Resource Structure of oneM2M .....	9
Fig. 3. Structure of Mobius Platform .....	13
Fig. 4. Topic based Information Architecture .....	16
Fig. 5. 5W1H based context classification .....	17
Fig. 6. Structure of context based dynamic access control model .....	18
Fig. 7. An FDBS and its components .....	20
Fig. 8. Taxonomy of multidatabase systems .....	21
Fig. 9. Automated process usecase for request .....	22
Fig. 10. Data identification of CUAHSI ODM .....	23
Fig. 11. CUAHSI ODM Overview .....	24
Fig. 12. Legacy device management technology and oneM2M service layer ..	25
Fig. 13. JBoss Data Virtualization .....	28
Fig. 14. Database integration system .....	29
Fig. 15. Structure of resource management system .....	38
Fig. 16. Data search processes based on 5W1H .....	40
Fig. 17. IoT data federation concept .....	43
Fig. 18. Basic statistics management schema .....	48
Fig. 19. Data store/retrieve structure for statistic analysis .....	49
Fig. 20. Dashboard of mandarin platform .....	50
Fig. 21. Things & sensors status UI of mandarin platform .....	51
Fig. 22. Things management UI of mandarin platform .....	52
Fig. 23. Sensors management UI of mandarin platform .....	52

Fig. 24. Sensor logs management UI of mandarin platform .....	53
Fig. 25. Table schema for sensor data .....	54
Fig. 26. Thing registration UI .....	55
Fig. 27. Measurement type management UI .....	56
Fig. 28. User friendly filter UI .....	57
Fig. 29. Database integration setting UI for legacy system DB .....	61
Fig. 30. Legacy heterogeneous systems federation management UI .....	62
Fig. 31. Structure of data federation agent .....	64
Fig. 32. UI for data analysis .....	65
Fig. 33. Matrix style dataset output .....	66
Fig. 34. CSV file of dataset .....	67
Fig. 35. R program code for basic analysis .....	68
Fig. 36. Flow of data translation test .....	71
Fig. 37. Union query settings .....	73
Fig. 38. Output of basic statistics calculation .....	76
Fig. 39. Box plot chart .....	77
Fig. 40. Time series plot chart .....	77
Fig. 41. Correlation plot chart .....	78
Fig. 42. Scatter plot chart .....	79
Fig. 43. Regression analysis .....	80

# 표 목 차

Table 1. Data quality standards of DQC .....	4
Table 2. Definition of "Data Usability" .....	5
Table 3. Filter Criteria conditions of oneM2M .....	10
Table 4. 5W1H expressive review of oneM2M Filter Criteria .....	11
Table 5. 5W1H and BDI .....	18
Table 6. 5W1H mapping results .....	36
Table 7. Union query example .....	45
Table 8. Basic statistics definitions .....	47
Table 9. Development and experiment environment of DFMS .....	58
Table 10. Heterogeneous system's information for federation .....	59
Table 11. Data translation testing overview .....	70
Table 12. Federated systems for union query .....	72
Table 13. Results of union queries .....	75
Table 14. Data integration and utilization procedures .....	82

## 데이터 활용성 강화를 위한 사물인터넷 서비스 플랫폼 확장

컴퓨터공학과 민 성 현  
지도교수 이 상 준

본 논문에서는 사물인터넷 서비스 플랫폼 설계 시 데이터 활용성을 강화하기 위해 고려해야 하는 기능 세 가지를 제안하였는데, 첫째, ‘육하원칙 기반의 IoT 데이터 검색 기능’, 둘째, ‘에이전트 기반 IoT 연합 DB 구성 기능’, 셋째, ‘IoT 데이터의 통계적 처리 기능’이 그것이다.

제안된 기능들에 대해서는 각각 필요성 및 설계 아이디어를 제시하였고, 이를 사물인터넷 플랫폼 상에서 동작할 수 있는 상태로 구현 후 실험을 수행하였다. 또한, 의도한대로 기능들이 수행되어 종래의 방식 대비 작업 절차 단축이 가능함을 확인하였다.

이렇게 제안된 기능들을 활용하여 종래의 커넥티비티 중심 플랫폼이나 디바이스 중심 플랫폼을 확장하면 데이터 활용성을 한층 강화할 수 있게 된다. 이는 수평적 플랫폼으로서의 비전을 보다 충실히 구현할 수 있음을 의미하며, 이로써 이종 도메인 간 데이터 공유 및 융합 서비스 개발의 가속화에 기여할 것으로 기대된다.

주제어 : 사물인터넷 서비스 플랫폼, 데이터 활용성, 육하원칙, 이종 시스템, 데이터 검색 규칙, 통계 분석, 데이터 통합, 서비스 융합, 연합 데이터베이스

Abstract

## An Expansion of IoT Service Platform to Enhance Data Usability

Seong-Hyeon Min  
Department of Computer Engineering  
Graduate School  
Jeju National University

Supervised by Professor Sang-Joon Lee

In this paper, we proposed three features that should be considered in order to enhance data usability in designing IoT service platform. First, 'IoT data retrieval feature based on 5W1H', second, 'Federated IoT database building feature based on agent', third, 'Statistical processing feature of IoT data'.

For each of the proposed features, the necessity and the design idea were presented, and the experiment was performed after implementing it in a state that it can operate on the IoT platform. In addition, it was confirmed that the features are performed as intended and the working procedure can be shortened compared with the conventional method.

Utilizing these proposed features, we can further enhance data usability by expanding the traditional connectivity-centric platform or device-centric platform. This means that the vision as a horizontal platform can be implemented more faithfully, which is expected to accelerate the data sharing and convergence service development between heterogeneous domains.



Keywords : IoT service platform, data usability, 5W1H, heterogeneous system, data search criteria, statistical analysis, data integration, service convergence, federated database

## Abbreviations

API	Application Programming Interface
BBF-TR	BroadBand Forum - Technical Report
CoAP	Constrained Application Protocol
CSV	Comma Separated Values
CUAHSI	Consortium of Universities for the Advancement of Hydrologic Science
DBMS	DataBase Management System
DFA	Data Federation Agent
DFMS	Data Federation Management System
DQC	Data Quality Certification
IoT	Internet of Things
MQTT	MQ Telemetry Transport
NoSQL	Not Only Structured Query Language
ODM	Observations Data Model
OMA DM	Open Mobile Alliance Device Management
RDBMS	Relational DataBase Management System
SaaS	Service As A Service
URI	Uniform Resource Identifier

# I. 서론

## 1. 연구 배경 및 목적

‘사물인터넷’ 또는 ‘IoT(Internet of Things)’라는 이름을 내세운 제품이나 서비스들이 스마트홈, 교통통제, 보안관리 등 다양한 분야에서 소개되고 있다[1-3]. 이처럼 구체적인 사물인터넷 서비스가 등장하고, 기술 생태계가 확장되어 감에 따라 ‘사물인터넷 플랫폼 기술’ 고도화에 대한 요구 또한 증대되고 있다.

특히, ‘서비스 플랫폼 기술’은 최근의 빅데이터 기술 확산 추세와 맞물리면서 이중도메인 간 자원의 공유와 서비스 융합에 대한 능력까지 요구받고 있는 실정이다. 이러한 흐름을 반영하듯 전 세계 주요 표준화 기관 및 기업들이 각 분야별로 컨소시엄을 구성하여 서비스 플랫폼 규격에 대한 표준 제정을 위해 노력하고 있다.

주목할 점은 수직적 구조에서 수평적 구조로의 아키텍처 패러다임 전환이 이루어지고 있다는 점이다. 기존의 시스템들은 도메인별로 구축되고 해당 도메인 내에서만 자원의 공유 및 활용이 이루어지는 수직적 서비스 구조였다. 반면, 최근의 기술 개발 방향은 하나의 공통 플랫폼을 통해 서로 다른 도메인 간에도 자원 공유 및 활용이 가능한 수평적 구조의 개방형 생태계 구현이라는 목표를 향해 지속적으로 발전해가고 있다[4]. Fig. 1은 oneM2M 표준[5]이 제시하는 수평적 플랫폼 비전을 표현하는 개념도이다.

현 시점에서 꼽을 수 있는 대표적 개방형 사물인터넷 플랫폼 기술로는 oneM2M[5], AllJoyn[6], IoTivity[7] 등이 있는데, 그 중 oneM2M 표준은 광역 네트워크를 포함할 수 있는 기술로서[8] 대규모 도메인 또는 다중 도메인에도 적용 가능하며, 이론적으로는 무한대로 규모 확장이 가능한 구조를 가지고 있다.

이러한 구조적 특성 때문에 oneM2M 표준 기반의 플랫폼은 AllJoyn, IoTivity 등 경량 도메인용 타 플랫폼 기술에 비해 상대적으로 더 다양하고, 많은 데이터를 취급할

수 있게 된다.

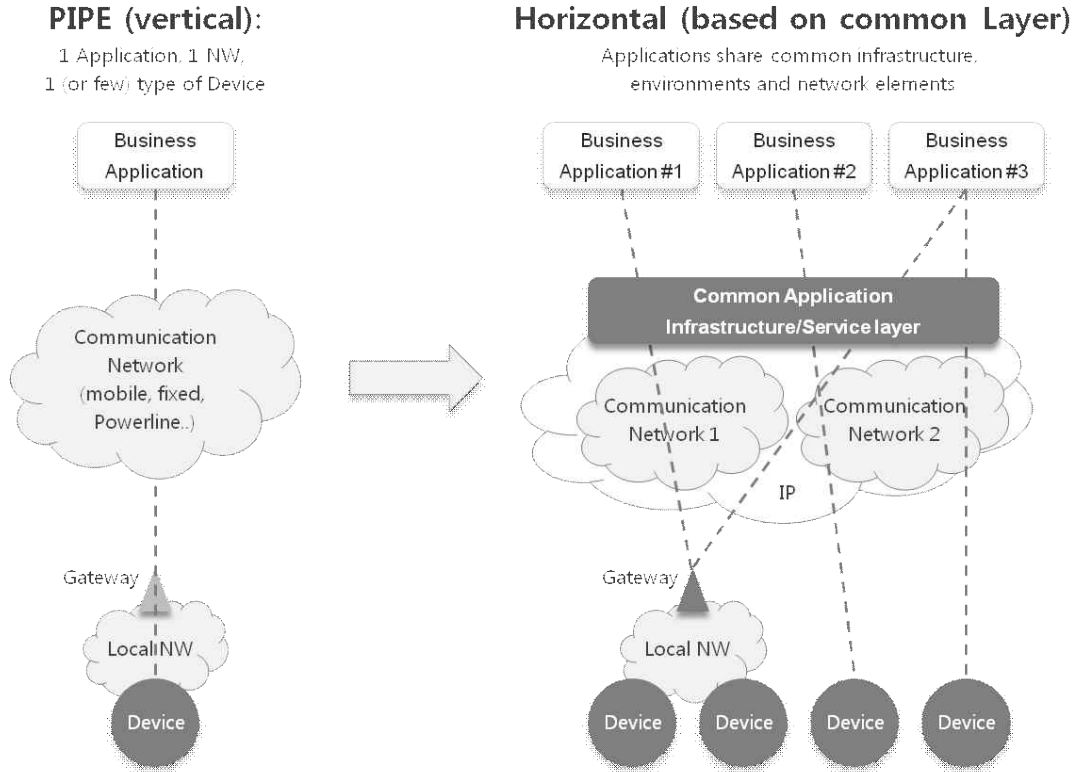


Fig. 1. Horizontal platform vision of oneM2M

따라서, oneM2M 표준과 같이 다양한 도메인에 대한 통합 기능을 제공하는 개방형 플랫폼을 설계할 때 수평적 플랫폼이라는 비전의 실질적 구현을 위해서는 데이터 활용성 강화와 관련 있는 기능들이 충분히 제공되어야만 한다. 사용자 관점에서는 원하는 데이터가 있는지 없는지, 있다면 어디에 있는지 알 수 없어서 사용하지 못하는 경우도 있고, 또는 데이터의 존재를 확인한 후에도, 그 활용을 위해 수행해야 하는 작업의 단계가 너무 많거나 또는 작업의 난이도가 너무 높은 경우 해당 데이터의 사용을 포기하게 되는 경우도 있을 수 있기 때문이다. 이렇듯 수평적 플랫폼 비전은 높은 수준의 데이터 활용성이 담보될 때에만 비로소 구현이 가능해진다.

그러나 기존 서비스 플랫폼 기술들의 설계 관점은 아직 장치 및 서비스 간 ‘연결성 (Connectivity)’ 확보에 큰 비중을 두는 단계에 머무르고 있으며, 데이터 활용성 강화에

대한 고려는 상대적으로 부족한 실정이다. 사물들이 서로 인터넷을 통해 연결되게 한다는 것이 사물인터넷의 핵심 개념이라는 점을 감안하면, 연결성 확보가 최우선적으로 고려되는 것은 필연적인 것으로서, 기술 발전을 통해 연결성 관련 이슈들이 해결되어감에 따라 자연스럽게 데이터 활용성에 대한 관심 수준도 높아질 것으로 예상된다.

그러나, 현 시점에서 공개되어 있는 기존 플랫폼들은 데이터 활용성 강화를 위한 확장이 필요할 것으로 판단되며, 이에 본 논문에서는 사물인터넷 서비스 플랫폼의 데이터 활용성 강화 방안을 연구한다.

데이터 활용성의 정의는 한국데이터진흥원의 데이터 품질 인증(이하 DQC)[9]이 개발한 데이터 품질관리 성숙모형에 제시된 데이터의 품질 기준을 참조하였다. Table 1은 DQC가 이 기준을 정리하여 제시한 표이다. 이 모형은 데이터 활용성을 유효성, 활용성으로 구분하여 정의하였으며, 이 중 활용성은 다시 접근성, 유용성, 적시성, 보안성 등 세부 특성으로 정의하였다.

본 논문에서는 사물인터넷 서비스 플랫폼의 데이터 활용성 강화를 위해 앞서 설명한 데이터 품질관리 성숙모형의 활용성 부문 세부 특성 네 가지 중 보안성을 제외한 세 가지 특성에 대해 각각 강화 방안을 도출한다.

접근성은 사용자가 원하는 데이터를 쉽게 검색할 수 있게 하는 특성을 포함하는 개념이다. 접근성 강화를 위해 사용자 친화적인 IoT 데이터 검색 기준을 제공하는 연구를 수행하였다. 사용자 친화적인 검색을 위해서 상황인지 분야에서 데이터 모델 설계 시 활용하는 육하원칙 기반 아키텍처를 참조하였다. 또한, IoT 생태계에서 활용하는 개념들을 육하원칙 요소들에 사상하는 작업을 수행하였다.

유용성은 사용자가 원하는 데이터의 범위와 상세화 정도에 대한 충족여부를 나타내는 개념이며, 유용성 강화를 위해서 에이전트 기반 연합 DB 구성 방안을 연구하였다. 이 과정에서 연합 DB(Federated DataBase) 시스템[10-13] 모형을 참조하여 이종 시스템의 데이터에 대해 통합 질의를 가능하게 하는 방법을 제안한다.

적시성은 적절한 응답시간 및 데이터 최신성 유지에 관한 품질 특성이다. 적시성 강화를 위해서는 IoT 데이터의 통계적 처리 방안을 연구하였다. 이는 원시데이터의 공유를 넘어서 통계적 활용을 위한 전처리 과정과 그 결과 데이터까지 공유할 수

있도록 하는 방법에 관한 것이다.

Table 1. Data quality standards of DQC

Data quality standards	Usability	Usefulness	Adequacy
			Flexibility
			Usability
			Traceability
		Accessibility	Accessibility
		Timeliness	Timeliness
	Security	Protective	
		Responsibility	
		Safety	
	Validity	Accuracy	Truth
			Suitability
			Essentiality
			Correlation
		Consistency	Compatibility
			Conformity
Integrity			

또한, 본 논문에서는 연구 주제와 관련 있는 기존 연구들에 대한 분석을 통해 서비스 플랫폼 설계 및 구현 시 데이터 활용성을 강화하기 위해 고려해야 할 요건들을 추가적으로 도출한 후, 설계 및 구현 과정에서 각 요건들을 적용하기 위한 방안에 대해서도 연구하였다.

데이터 활용성과 관련해서는 박고은, 김창재[14]의 연구도 참조하였다. 해당 연구는 데이터 활용성을 ‘자유로운 가공을 통한 가치 재창출을 위해 데이터가 갖춰야 할 특성’으로 정의하였으며, 기존 연구들을 바탕으로 데이터 활용성을 다섯 가지 세부 특성으로 설명하고 있다. 박고은, 김창재[14]는 이 특성들을 Table 2와 같이 정리하였다.

Table 2. Definition of "Data Usability"

Usability	Definition	References
	Adequate for value creation through processing or re-using the open data	
Undertandability	· Understandable for various users without confusion or distortion	ISO 25012 (Understandability)
Processability	· No limitations for data processing, such as hardware or software limitations. · No need for special support or configurations for using data	DQI (Flexibility), ISO 25012 (Accessibility)
Linkage	· The leverage is available in variety environments and situations. Also, can be used in conjunction with other data(different from context, systems, institutions, and countries) · Redundant data inside the institution or associated data have a consistent management standard	DQI (Flexibility)
Convenience	· Data can be provided for users with a familiar format · User satisfaction is high	DQA (Utility)
Compliance	· Compliance with the administrative standard terms and codes · Easy for Integrated government data linkage and common use of administrative information · Compliance with government data standards and institution data standards	Public Data Management Guidelines (Readlines), ISO25012 (Compliance)

본 논문에서는 제안하는 설계와 구현 과정에서 이들의 연구가 제시하는 데이터 품질 특성들 중 ‘연계성’, ‘친숙성’, ‘표준 준수성’ 등의 향상을 위한 방법 또한 고려하였으며, 이러한 특성들은 본 논문에서 제안하는 세 가지 기능에 수반되는 세부 요구사항으로 반영하였다.

이러한 과정을 통해 데이터 활용성이 강화된 플랫폼에서 작업을 수행하면 기존 기술 대비 작업 절차는 단축되고, 작업의 난이도는 수용 가능한 수준으로 낮추는 효과를 기대할 수 있다.



## 2. 논문의 구성

본 논문에서는 기존 사물인터넷 서비스 플랫폼들이 데이터 활용성 강화를 위한 기능들을 충분히 제공하지 못하고 있는 문제를 해결하고자 세 가지 확장 기능을 제안하였다.

II장에서는 사물인터넷 데이터 검색, 이중 시스템 데이터 통합에 관한 기존 연구들에 대해 분석한 결과를 소개한다. 또한, 각 연구별로 도출한 문제점이나 제약사항들을 제시하고, 그에 대한 해결 아이디어로서 도출한 내용 및 기존 연구에서 참조하고자 하는 내용들도 함께 제시하였다.

III장과 IV장에서는 데이터 활용성 강화를 위해 고려해야 할 요건 세 가지에 대해 각각 설계 및 구현 결과를 기술하였다. 먼저, 육하원칙을 사물인터넷 개념요소에 사상한 후, 이를 oneM2M 표준의 <Filter Criteria>에 적용하기 위한 확장안과 시스템 구성안을 제시하였다. 또한, 육하원칙 각 질문요소들을 포함하는 쿼리들이 본 연구에서 제안하는 시스템 구성 안에서 수행되는 절차를 설명하였다.

이어서, 사물인터넷 서비스 플랫폼이 통계적 분석에 적합한 구조로 데이터를 제공하는 방법과 플랫폼 사용자들이 축적된 데이터를 이용하여 탐험적 분석을 수행할 수 있게 하는 방법을 제시하였다.

또한, 이중 시스템 데이터를 서비스 플랫폼에 통합시키고자 할 때 그 과정을 효과적으로 단축시키면서도, 레거시 시스템의 운영에 미치는 영향을 최소화하는 방법에 관해 설명하고, 설계 및 구현 결과를 제시하였다.

V장에서는 IV장에서 제시한 구현결과를 바탕으로 기능 시험을 수행한 결과를 제시하고, 데이터 활용성 향상을 통한 작업 절차 단축 가능성에 대해 평가하였다. VI장에서는 결론을 맺고, 향후 추가로 진행되어야 할 연구에 대해 제안하였다.

## II. 관련 연구

### 1. 사물인터넷 데이터 검색

#### 1) oneM2M 표준의 자원 발견 방법

사물인터넷 서비스 플랫폼 표준을 개발하기 위해 2012년 출범한 oneM2M[5]은 2015년 1월 릴리즈 1을 공개한 이후 지속적인 활동을 통해 자료의 갱신을 거듭하고 있으며, 2016년 8월에는 릴리즈 2를 공개한 바 있다. 또한, 각 릴리즈별로 다양한 규격정의 문서들이 개발되어 있는 상태다.

이 규격서들 중 플랫폼 설계에 관한 내용을 정의하고 있는 문서인 Architecture(TS-0001-V1.13.1)[15]에 따르면, oneM2M 표준은 플랫폼 내 모든 자원들을 <Resource>라는 단위로 관리하는데, 그 중 IoT 디바이스를 통해 수집되는 ‘측정데이터’의 경우 <container>와 <contentInstance>라는 <Resource Type>을 통해 구조화하며, 이를 계층구조로 관리한다. <container>와 <contentInstance>는 Fig. 2와 같이 각각 파일시스템에서의 디렉토리나 파일의 개념처럼 사용되며, 측정데이터 역시 계층구조 하에서 지정된 경로에 저장된다. 이 때, 하나의 측정데이터는 하나의 <contentInstance>로 취급된다.

상호운용성 관점에서 중요한 것은 데이터 획득방식인데, oneM2M 표준에서 자원의 접근은 계층구조를 표현하는 URI(Uniform Resource Identifier)를 통해 이뤄진다. 그러나, 이러한 방식은 가져오고자 하는 자원의 자원명과 해당 자원이 위치한 계층구조의 URI를 정확하게 알고 있는 경우가 아니라면 접근이 어렵다는 단점이 있다. 특히, 할당받은 공간 내에서 다중으로 <container>를 생성하는 식으로 자유롭게 계층구조를 관리할 수 있다는 점을 고려하면 이중 도메인 간 데이터 교환 시 해당 도메인의 계층구조에 대한 학습 없이는 자원 접근이 사실상 불가능하다고 할 수 있다.

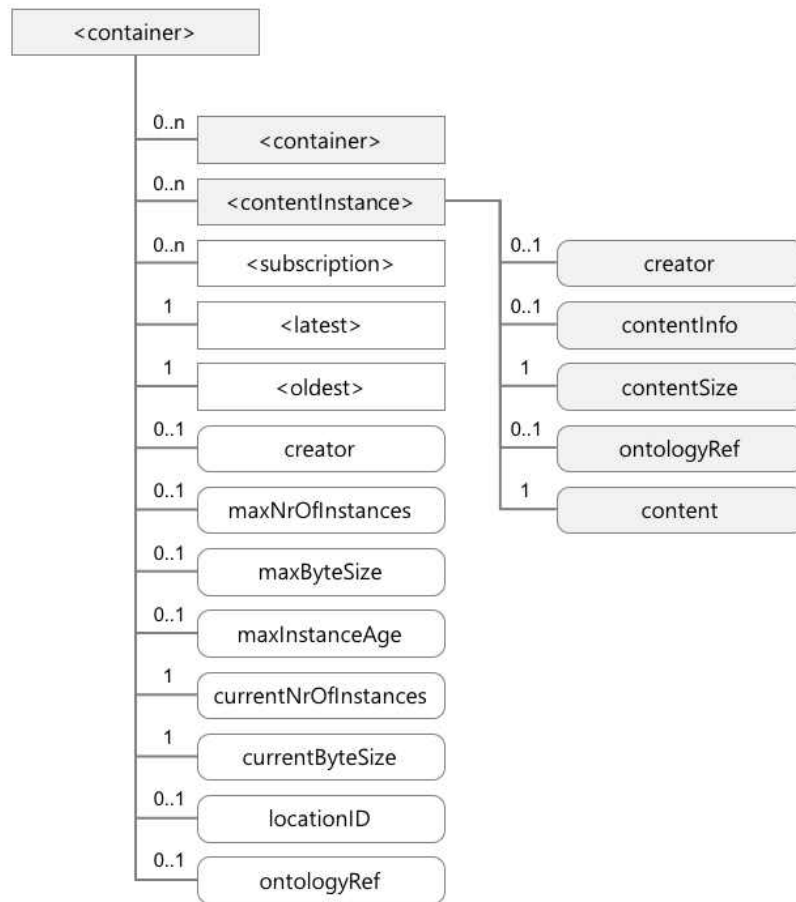


Fig. 2. Resource Structure of oneM2M

다른 방식으로는 Table 3과 같은 <Filter Criteria>를 이용하여 데이터를 검색하는 방법이 있다. 검색하고자 하는 경로 URI에 검색조건에 해당되는 파라미터를 붙여서 질의하는 방식이다. 이는 자원의 전체경로를 입력하는 방식에 비해 포괄적으로 유연하게 데이터를 검색할 수 있다는 장점이 있다.

조건을 통해 원하는 데이터를 정확하게 검색하기 위해서는 가용한 <Filter Criteria>가 충분한 표현력을 갖추어야 한다. 검색하고자 하는 측정데이터들은 그 각각이 독립된 하나의 상황에 대한 기록이라고 할 수 있으므로, 상황을 설명할 수 있는 다양한 요소들을 조건으로 사용할 수 있어야 한다. 일반적으로 어떠한 상황을 완전하게 설명하기

위해서는 육하원칙을 사용한다[16]. 단일 도메인 내에서 데이터 모델을 설계할 때에는 해당 도메인 이해관계자들이 같은 지식체계를 공유하고 있는 상태를 가정하기 때문에 육하원칙을 엄격하게 적용할 필요가 없었지만, 다양한 도메인에서 공유할 데이터 모델이라면 특정 도메인의 지식체계를 알지 못하더라도 상황에 대한 표현만으로 데이터 검색이 가능하도록 육하원칙 기반의 <Filter Criteria>를 제공할 수 있어야 할 것이다.

Table 3. Filter Criteria conditions of oneM2M

Condition tag	Multiplicity	Matching condition
<i>createdBefore</i>	0..1	Resource generation time is before
<i>createdAfter</i>	0..1	Resource generation time is after
<i>modifiedSince</i>	0..1	Resource update time is before
<i>unmodifiedSince</i>	0..1	Resource update time is after
<i>stateTagSmaller</i>	0..1	updated version is Smaller than the number specified
<i>stateTagBigger</i>	0..1	updated version is Bigger than the number specified
<i>expireBefore</i>	0..1	Resource Expiration time is before
<i>expireAfter</i>	0..1	Resource Expiration time is after
<i>labels</i>	0..n	Label equal to the specified value
<i>resourceType</i>	0..n	Resource type equal to the specified value
<i>sizeAbove</i>	0..1	Resource size is above the specified value
<i>sizeBelow</i>	0..1	Resource size is bellow the specified value
<i>contentType</i>	0..n	Resource information attribute equal to the specified value
<i>limit</i>	0..1	Results Return limited number
<i>attribute</i>	0..n	Resource attributes equal to the specified value(ex:creator)
<i>filterUsage</i>	0..1	Filter usage equal to the specified value(ex:discovery)

oneM2M 표준에서 제공하는 <Filter Criteria>는 이러한 관점에서 검토한 결과, Table 4와 같이 표현력이 미흡하거나, 효율적이지 않은 점들을 발견할 수 있었다.

제시된 조건요소들 중 대부분은 <업데이트 버전 (stateTagSmaller / stateTagBigger)>, <만료일자 (expireBefore / expireAfter)>, <용량 (sizeAbove / sizeBelow)> 등 자원의 상태 표현이나 시스템 관리에 필요한 요소들이고, 측정 데이터가 표현하고자 하는 상황을 설명할 수 있는 육하원칙 만족요소는 <생성일시 (createdBefore / createdAfter)> 요소와 <속성(attribute)>으로서의 <creator> 요소가 각각 '언제'와 '누가'에 대한 조건으로 사용될 수 있을 뿐 '어디서', '무엇을', '어떻게', '왜'에 대해서는 조건으로 사용할 수 있는 방법이 없다.

Table 4. 5W1H expressive review of oneM2M Filter Criteria

5W1H	oneM2M Filter Criteria	Expressiveness	Description
When	createdBefore, createdAfter	O	can be used unchanged.
Who	creator of attribute	△	A device ID to generate the data can only be viewed. But can not know who is the agent of the data collection practices.
What	label, resourceName of attribute	△	It can deduced what the data is, but that is not clear
How	creator of attribute	X	A device ID to generate the data can only be viewed. But can not know the data collection system or device type.
Why	N/A	X	not considered.
Where	N/A	X	It provided with a process to obtain the location information of a specific device within the platform, but can not be used as a Filter Criteria.

이에, 본 논문에서는 oneM2M 표준의 Filter Criteria가 표현하지 못하는 나머지 육하원칙 요소들도 자원 발견 조건으로 활용할 수 있도록 확장된 자원관리 모델을 연구한다.

## 2) Mobius Platform의 Global Discovery Platform

최성찬 등[17]에 따르면 Mobius 플랫폼은 oneM2M 표준을 실제 운용 가능한 서버 플랫폼으로 구현한 제품이다. oneM2M에서 정의하고 있는 공통 서비스 기능들을 포함하고 있으며, HTTP, CoAP[18], MQTT[19] 프로토콜을 지원한다. 또한, 개발된 기술을 오픈소스형태로 공개하고 있으며, Fig. 3이 그 구성도이다. 또한, 점선으로 표시된 부분은 이미 공개된 기술이다.

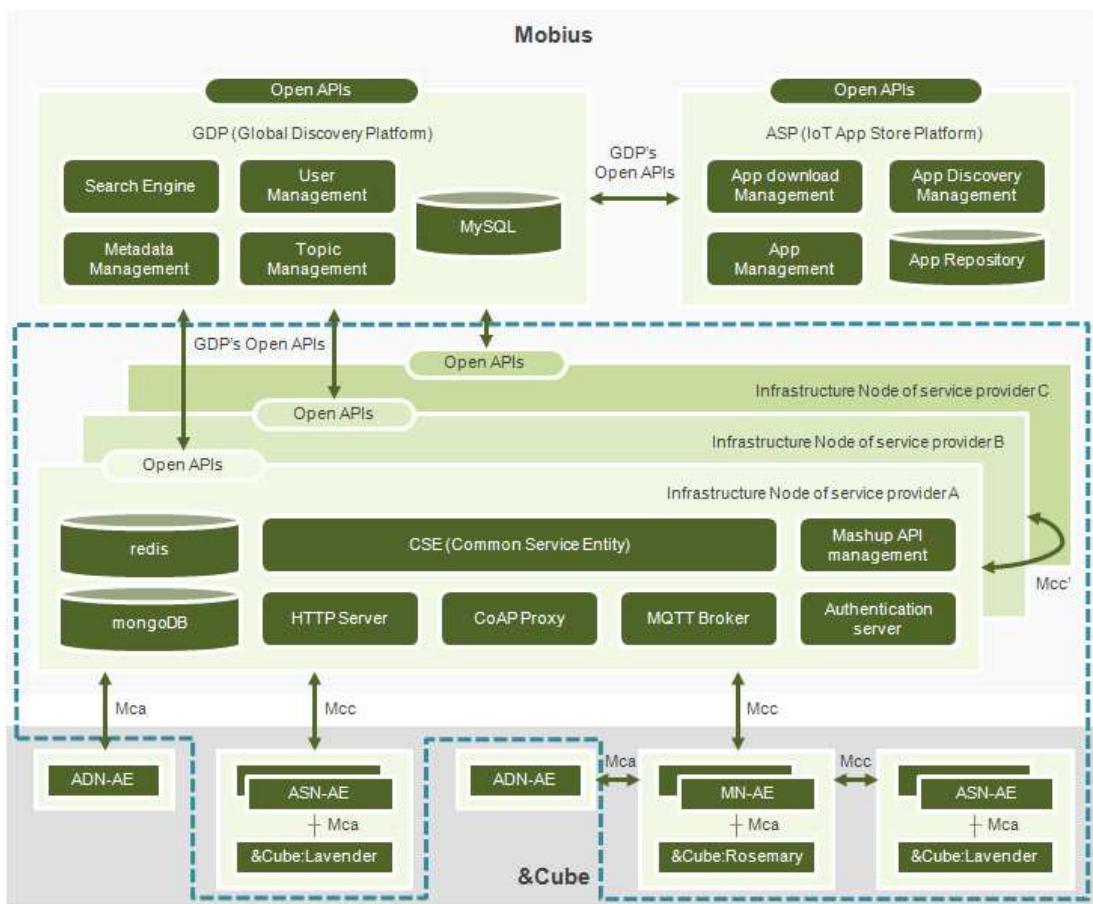


Fig. 3. Structure of Mobius Platform

Mobius 플랫폼의 특이한 점은 Fig. 3과 같이 자원 검색 기능을 확장하기 위해 GDP(Global Discovery Platform)를 추가 구현하였다는 것이다. Mobius 플랫폼은 이 GDP를 이용하여 키워드 기반, 위치기반, 디바이스 ID 기반 등의 글로벌 검색

을 지원하며, 자원의 그룹화를 지원하는 토픽 관리 기능을 포함하여 oneM2M 표준에서 제시된 내용 이상으로 확장된 방식의 데이터 획득 기능을 제공한다. 이는 oneM2M 표준 기반이 정의하고 있는 자원 발견규칙의 제약을 해결하기 위해 표준을 확장하는 기능으로, 사물인터넷 서비스 플랫폼에서의 자원 발견기능 보완 대한 필요성을 시사한다.



### 3) oneM2M 표준 자원 접근방식 확장

김지현 등[20]은 oneM2M 표준을 두 가지 관점에서 확장하여 사물인터넷 플랫폼 차별화 기능을 개발하는 방법을 제안하였다. 실무적 개발 과정에서 도출되는 확장 개발 요구사항들을 표준에 포함시키기보다는 개별적으로 최적화해야 하는 대상으로 인식하고 수행한 연구이다.

해당 연구에서 제안한 두 가지 확장안 중 본 논문에서 소개하고자 하는 내용은 자원 접근방식에 관한 것이다. 디바이스 소유자가 아닌 경우에 접근하기 어려운 현재 구조를 개선하기 위하여 토픽(Topic)이라는 개념을 도입하고, 이 개념을 중심으로 새로운 정보구조가 생성되도록 함으로써, 서비스 플랫폼들이 지향하는 도메인 간 정보 융합, 메쉬업(Mashup) 방식의 서비스 개발이 용이하게 하였다. 이러한 정보구조를 이용하면, 사용자는 원하는 토픽을 기준으로 IoT 디바이스의 권한 획득 및 정보 공유를 수행할 수 있게 된다.

이 때, 이 연구에서 제시하는 Topic의 개념은 기온, 습도와 같은 데이터 유형을 의미한다. 그러나, 제시된 방식에 따르면, 데이터 소스 소유자가 자유롭게 Topic을 생성할 수 있으므로, Topic 취급관점의 일관성을 보장할 수 없고, 동일한 개념에 대해서도 여러 유의어가 혼용될 수 있으므로, 데이터의 파편화가 발생할 수 있다. 이에, 본 논문에서 제시하는 자원관리 모델에서는 이러한 문제점을 방지할 수 있도록 Topic에 대응되는 개념인 측정유형 속성을 메타데이터로서 취급하는 방식을 사용한다.

Fig. 4는 김지현 등[20]이 제시한 자원관리 모델의 개념도이다.

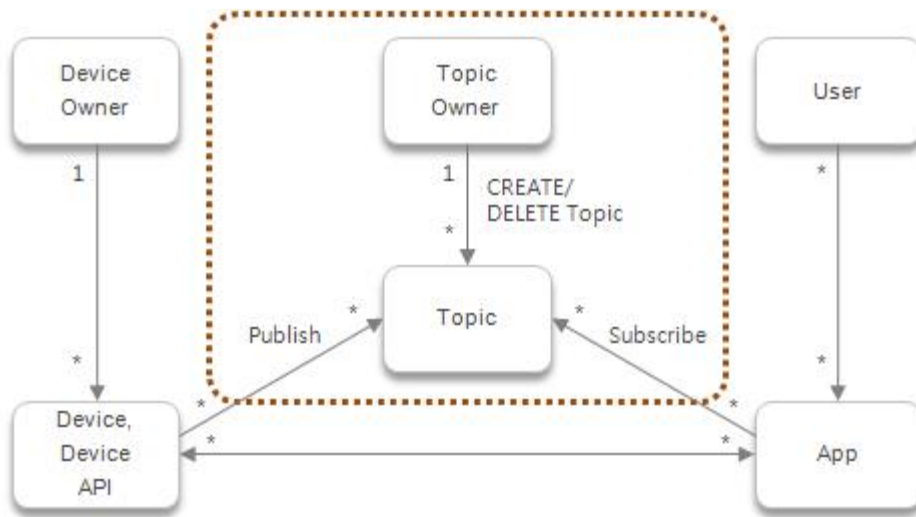


Fig. 4. Topic based Information Architecture

#### 4) 육하원칙 기반 상황인지 데이터 모델

2000년대 유비쿼터스 시스템에 대한 연구가 활발하게 이뤄지면서 컨텍스트에 대한 데이터 모델도 지속적으로 제안되었다. 송재구, 김석수[21]는 기존 방식이 상황에 따라 다른 표현을 취급하지 못하는 제약을 지적하고, 육하원칙 기반의 데이터 모델을 제안하면서 “5W1H는 사용자가 처한 상황의 정보를 포괄적으로 수용할 수 있으며, 정보 분류 및 조합에 유리하다.”고 하였다. Fig. 5는 이들이 제시한 육하원칙 기반 상황 분류 개념도이다.

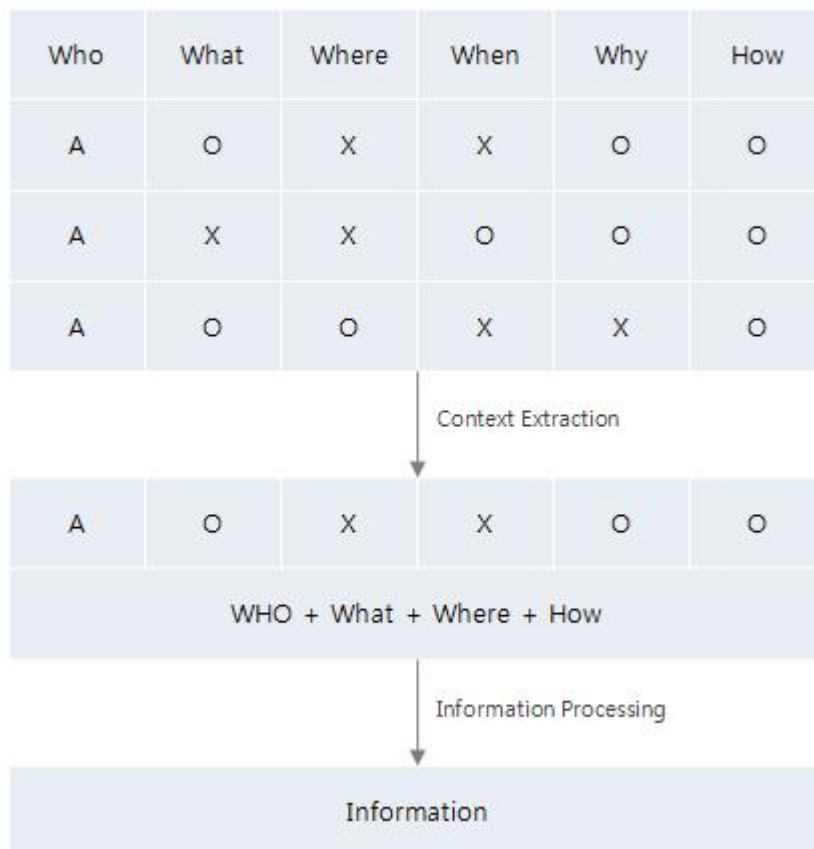


Fig. 5. 5W1H based context classification

이와 유사한 맥락에서 J. C. Augusto, J. O'Donoghue[22]는 상황인지 에이전트를 개발할 때 육하원칙 아키텍처를 활용할 것을 제안했다. 해당연구에서는 당시

많은 소프트웨어 에이전트 아키텍처들이 활용되던 BDI(Beliefs, Desires and Intentions) 모델을 구체화하기 위해 BDI 모델[23] 요소와 육하원칙 요소들에 대한 사상관계를 Table 5와 같이 제시하였다.

Table 5. 5W1H and BDI

5W1H	Concept	BDI
Who	Profile (including Needs / Preferences)	Beliefs
Where	Spatial conditions	
When	Temporal conditions	
Why	General aims	Desires
What	Specific goals	Intentions
hoW	Selection of plans	

최근에는 손지성[24]이 사물인터넷 자원에 대한 컨텍스트 기반 동적 접근제어 모델을 개발하면서 컨텍스트 관리자 구성시 육하원칙을 사용하는 온톨로지를 활용하는 방법을 제안하였다.

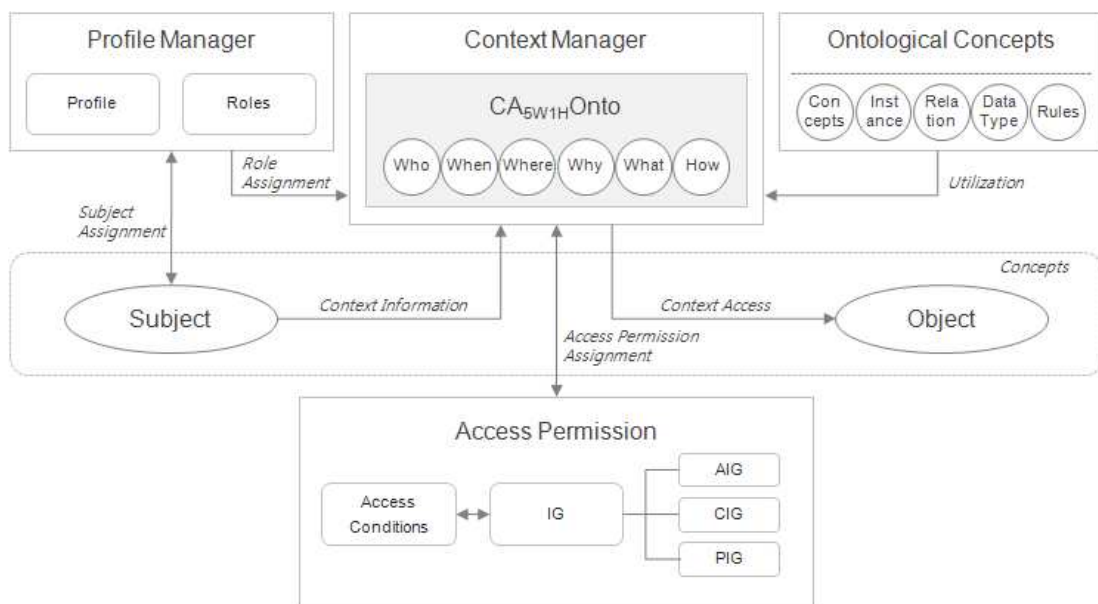


Fig. 6. Structure of context based dynamic access control model

Fig. 6은 손지성[24]이 제시한 상황 기반 동적 접근 제어 모델의 개념도이다.

이렇듯 컨텍스트 정보를 사용하는 연구들이 공통적으로 육하원칙을 활용했던 이유는 육하원칙이 상황설명에 대한 완전성 및 사용자 친숙성을 확보하기 쉽다는 장점 때문인 것으로 판단된다. 본 논문에서도 특정 도메인에 대한 지식 없이도 일반적으로 사용하는 어휘나 시간, 좌표 등의 조건들을 활용하여 원하는 데이터를 쉽고 정확하게 검색할 수 있도록 하기 위해서 육하원칙을 활용하는 데이터 검색 방법을 제시한다.

## 2. 이종 시스템 데이터 통합

### 1) 연합 DB 시스템

연합 DB 시스템(Federated DataBase System)은 자치성이 있는 다수의 DB를 하나의 연합 시스템으로 다룰 수 있게 하는 데이터베이스 관리 시스템의 일종이다[13]. 1985년 D. Heimbigner, D. McLeod[10]가 제안한 이후로 다양한 연구가 이루어졌으며, 데이터베이스 통합에 관한 다른 기법들과 다른 점은 관리의 자치성을 유지시키면서 인터페이스만 통합한다는 것이다.

A.P. Sheth, J.A. Larson[11]은 연합 데이터베이스 시스템에 대한 조사, 분석을 통해 설계 및 구현 시 고려해야 하는 이슈들과 해결방안을 정리했는데, 이들이 제시한 연합 DB 시스템의 구성은 Fig. 7과 같다.

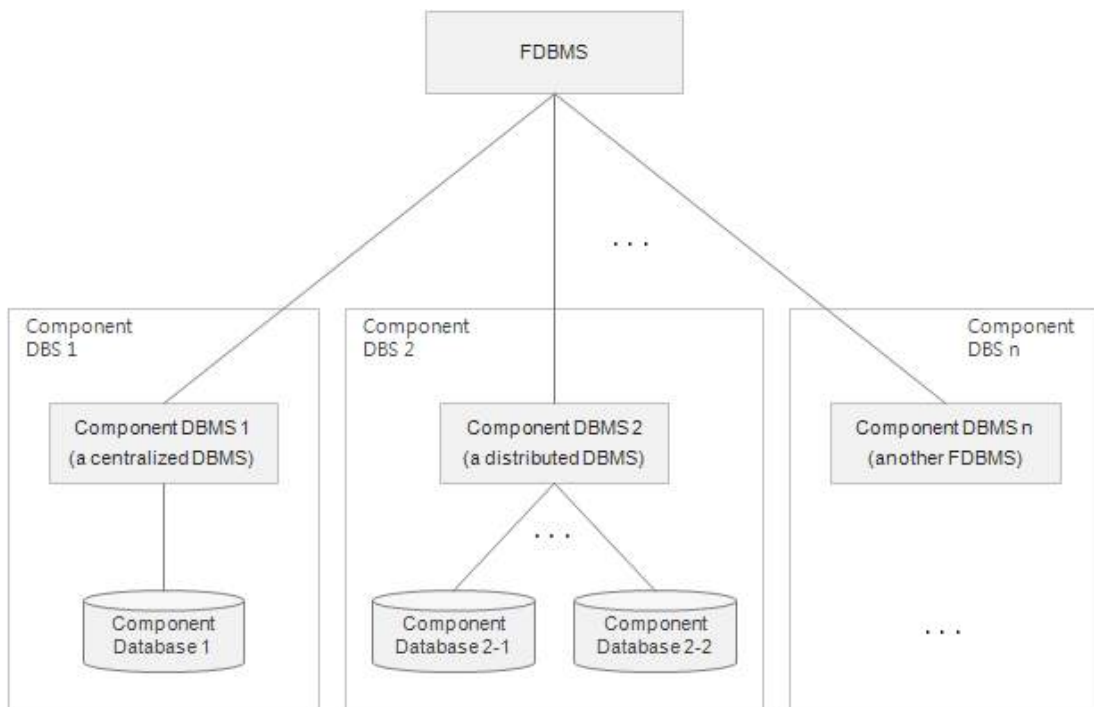


Fig. 7. An FDBS and its components

또한, 해당 연구에서는 연합 데이터베이스 시스템들을 Fig. 8과 같이 분류하였는데, 결합유형을 기준으로 Loosely Coupled, Tightly Coupled 시스템으로 먼저 구분하고, Tightly Coupled 시스템을 다시 Single Federation과 Multiple Federation으로 구분하였다.

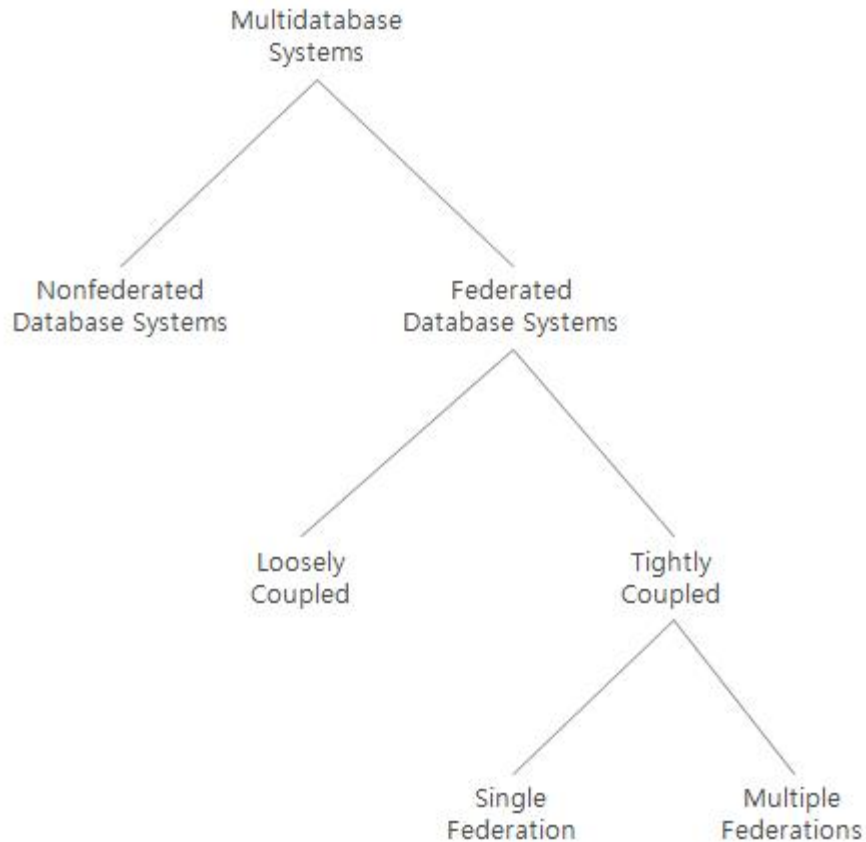


Fig. 8. Taxonomy of multidatabase systems

최근 김윤곤[12] 등은 연합 데이터베이스 시스템의 개념을 바탕으로 Fig. 9와 같이 스토리지 가상화 및 데이터 가상화를 통해 상호 호환성이 없는 이종의 스토리지에 대해 연합을 구성하고 데이터 중복을 줄일 수 있는 모델을 제안하였다.

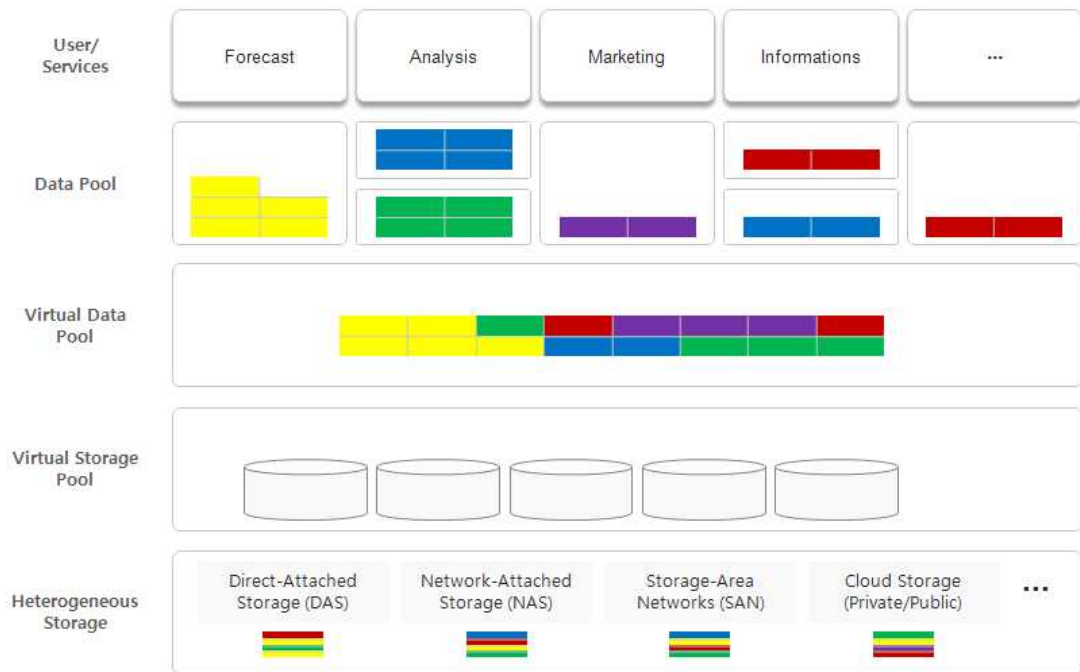


Fig. 9. Automated process usecase for request

이러한 연합 데이터베이스 시스템의 모델링 아이디어를 활용하여 본 논문에서는 이종 시스템의 데이터를 사물인터넷 서비스 플랫폼에서 공유할 수 있게 하는 에이전트 기반의 연합 구성 기술을 제안하였다.



## 2) CUAHSI 관측데이터 통합 모델

수문학(hydrology) 연구단체인 CUAHSI(The Consortium of Universities for the Advancement of Hydrologic Science, Inc.)[25]는 수문 연구용으로 수집되는 수많은 이종 데이터를 효과적으로 활용할 수 있도록 ODM(Observations Data Model)[26]이라는 수문 연구용 통합 데이터 모델을 개발했다.

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. 10은 관측 데이터를 취급하는 ODM의 관점을 표현하고 있다.

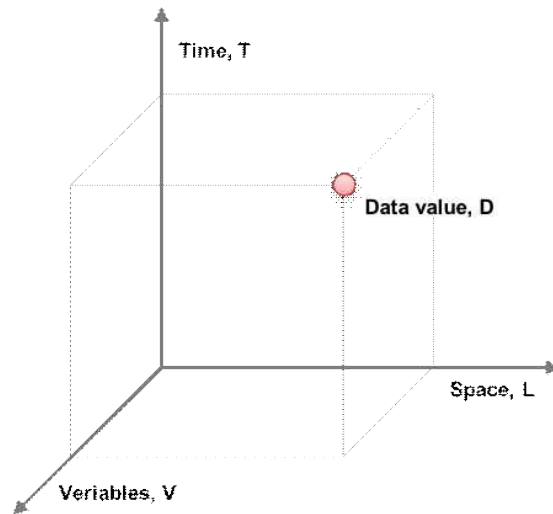


Fig. 10. Data identification of CUAHSI ODM

측정된 값 외의 다른 부가 정보들도 Fig. 11에서 보이는 것과 같이 연관된 테이블에 저장되므로 사용자는 필요에 따라 원하는 조건을 만족하는 값들을 용이하게 조회할 수 있으며, 관측 장치나 시스템 종류와 상관없이 특정 시간과 공간 범위를 조건으로 하여 모든 값들을 가져와서 분석할 수 있다는 점이 가장 큰 효용이라고 할 수 있다.

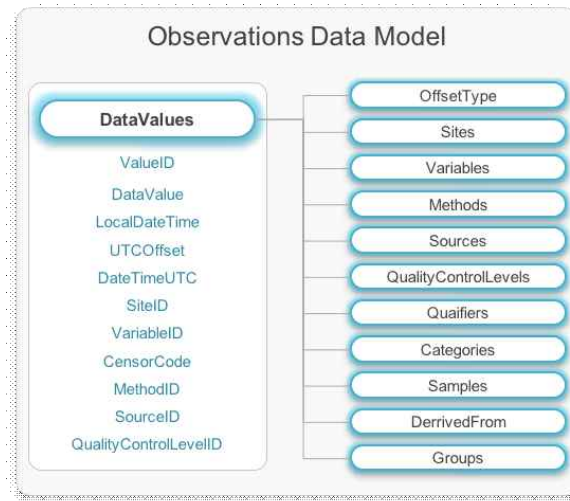


Fig. 11. CUAHSI ODM Overview

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

### 3) oneM2M 표준의 레거시 시스템 연동

사물인터넷 서비스 플랫폼의 구축 및 운영에 관한 기술표준인 oneM2M[5] 표준은 대형 도메인 및 다중 도메인에서의 수평적 서비스 통합을 지원하도록 개발되었다. 때문에, 레거시 시스템에 대한 연동 방안을 필수적으로 갖추어야 한다.

oneM2M 표준은 M2M 장치 관리 기술을 직접 개발하는 대신, OMA DM[27], BBF TR-069[28], OMA Lightweight M2M[29]과 같이 이미 상용화된 완성도 높은 기존 기술들을 선정하여 플랫폼이 이를 지원할 수 있게 하는 전략을 제시하였다[30]. Fig. 12는 oneM2M이 제시한 개념도로서, 서비스 레이어별로 연동 가능한 기존 장치 관리 기술들을 나타낸다.

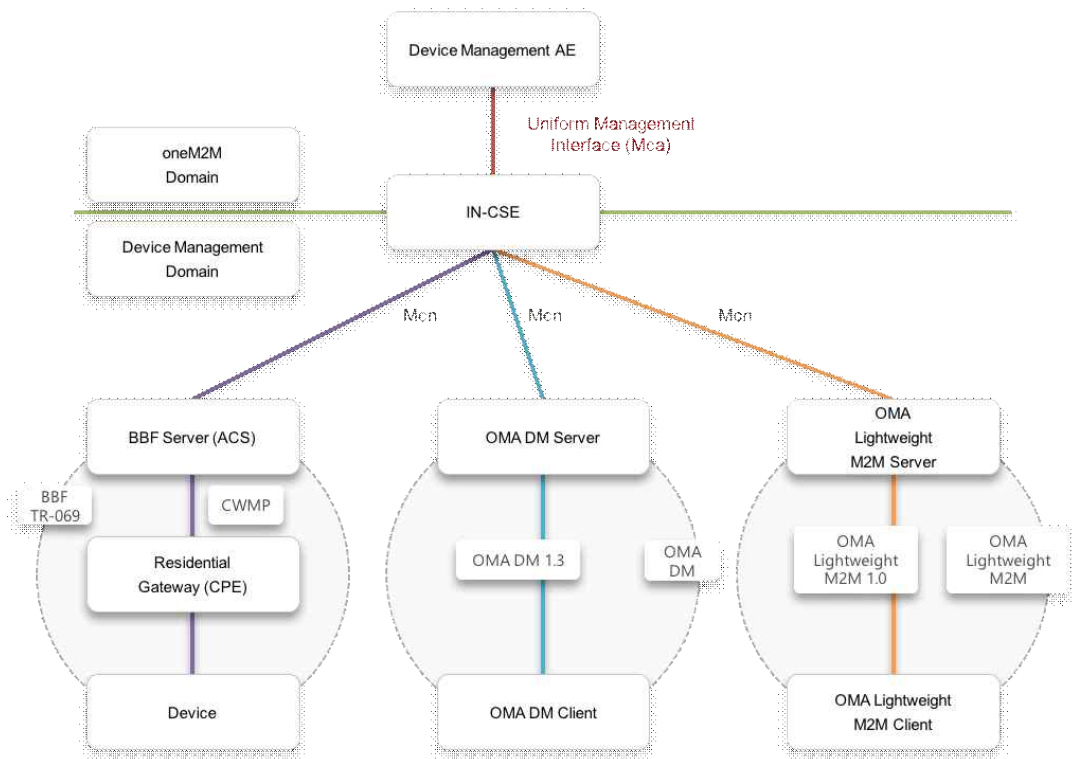


Fig. 12. Legacy device management technology and oneM2M service layer

그러나, 대형 시스템의 경우와는 달리 중소형 규모의 레거시 시스템들은 oneM2M 표준에서 지원 대상으로 선정한 기존의 기술표준도 지원하지 않는 경

우가 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.

oneM2M은 그러한 시스템들의 데이터를 활용할 수 있는 방법을 제공하지 않는다. 본 논문에서는 이런 경우 장치를 직접 관리할 수 없더라도 해당 시스템이 관리하는 DBMS와의 연동을 통해 제한적으로나마 해당 데이터소스를 활용할 수 있는 방안을 제시한다.

#### 4) 가상화 기반의 데이터 통합

JBoss Data Virtualization[31]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. 13와 같이 RDBMS, NoSQL[32], File, SaaS[33], Hadoop[34] 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 사용하고 자 하는 Data View의 스키마를 정의하고 레거시 데이터 소스와의 사상 규칙을 개발하는 하는 과정이 필요한데, 이 과정에서 분석 및 설계에 대한 노력이 요구된다. 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 사상 규칙을 제공함으로써, 관측데이터를 주로 활용하는 사물인터넷 서비스 플랫폼 구축 시 보다 간편하게 통합을 구현할 수 있도록 하는 데 연구의 초점을 맞춘다.

Fig. 13은 JBoss Data Virtualization이 제시하는 데이터 가상통합 솔루션의 개념도이다.

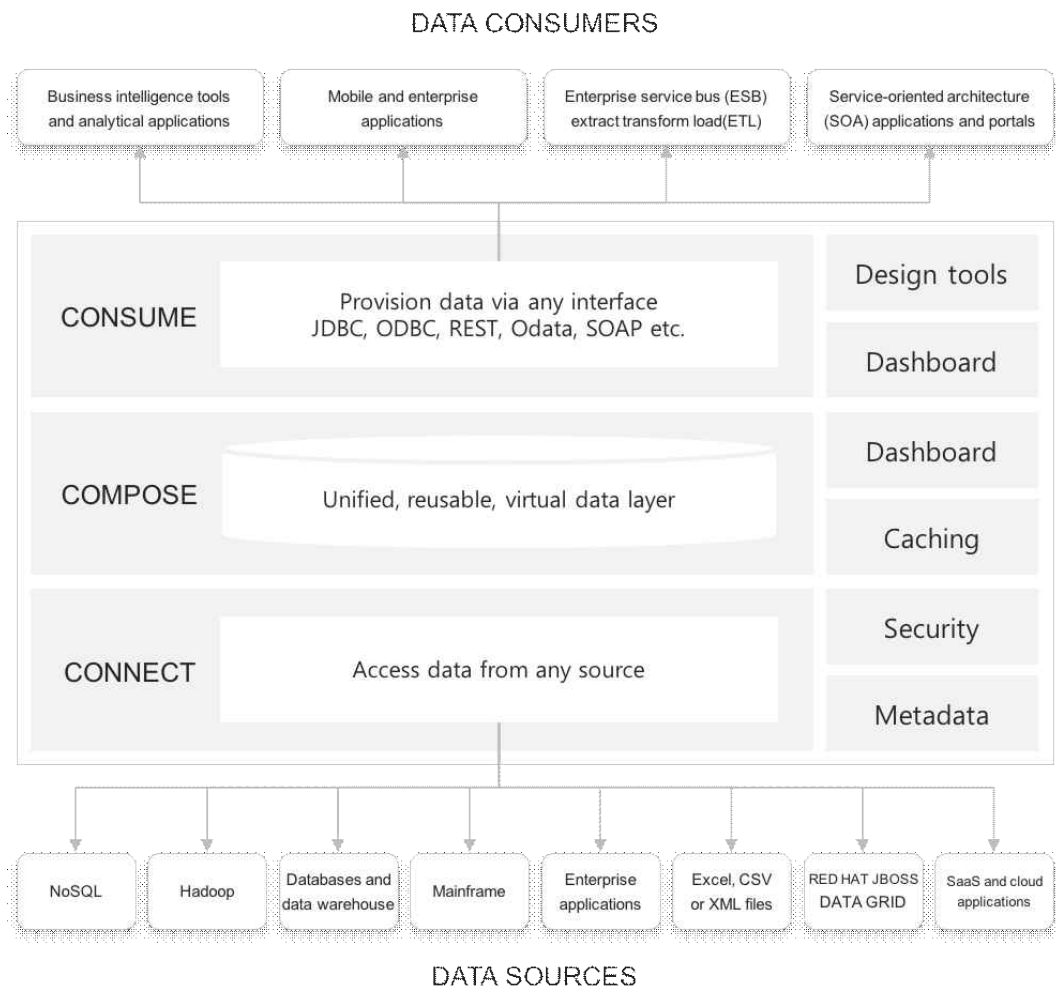


Fig. 13. JBoss Data Virtualization

## 5) 데이터 통합을 위한 전역 스키마 생성 기법

박우창[35]은 데이터베이스 연합 방식을 개선하여, 데이터베이스의 의미를 쉽고 효율적으로 찾아내는 통합 시스템을 제안하였다. 이를 위해, 전역 뷰(Global View)를 쉽게 생성할 수 있게 하는 도구를 개발하여 데이터베이스의 의미 정보 통합 기능을 구현하였다. 이 때, 이종의 DBMS를 취급할 수 있도록 UnityJDBC[36]를 이용하여, 각 로컬 데이터 소스에 대한 질의와 결과 추출이 이루어지도록 구현하였다. Fig. 14는 박우창[35]이 제시한 데이터 통합 시스템의 구성도이다.

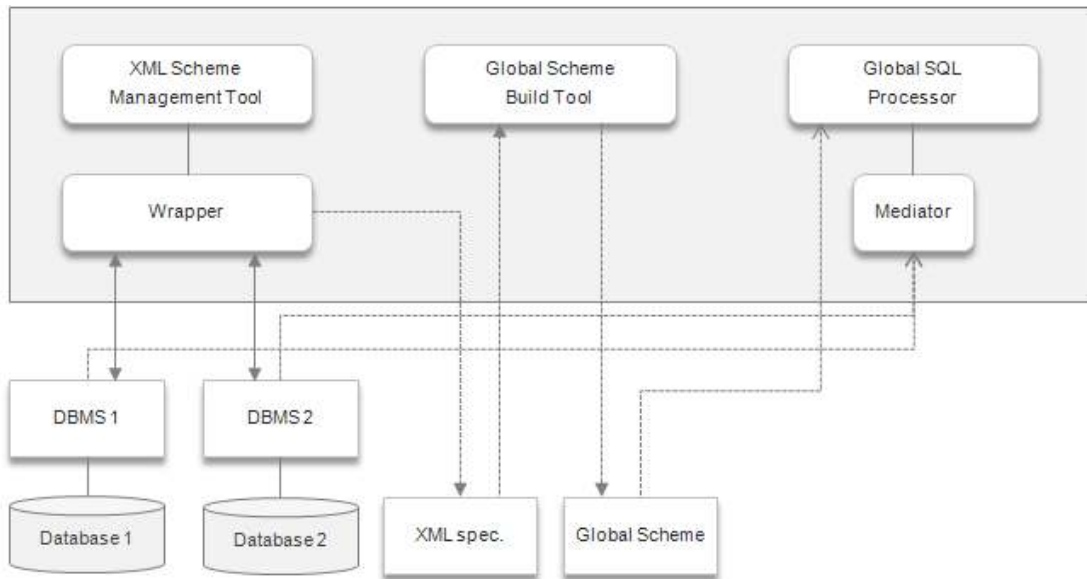


Fig. 14. Database integration system

이 제안에 따르면, 전역 뷰를 이용함으로써 로컬 데이터 소스의 구조에 대한 지식이 없더라도 사전에 정의한 전역 뷰의 정보만을 이용하여 질의를 수행할 수 있게 된다.

그러나, 이 구조에서는 시간형식, 값의 단위 등과 같은 각 데이터 속성의 형식적 의미적 통일에 대한 고려가 표현되어 있지 않다. 다수의 데이터베이스를 통합하는 작업은 질의 인터페이스를 통일하는 것뿐만 아니라, 별도의 정제 과정 없이

응용서비스 개발에 사용할 수 있도록 각각의 데이터 속성들까지 완전히 일관된 형식으로 통일시키는 작업도 필요로 한다. 따라서 본 논문에서는 이러한 부분까지 고려한 데이터 통합이 가능한 모델을 연구한다.



### III. 서비스 플랫폼 확장 설계

본 장에서는 사물인터넷 서비스 플랫폼의 기능 확장을 제안한다. 기능 확장의 목표는 데이터 활용성 강화이며, 본 논문에서 다루는 데이터 활용성에 관한 정의는 한국데이터진흥원에서 개발한 데이터품질기준[9]을 참조하였다.

제안된 확장 기능은 총 세 가지인데, 이 중 두 가지는 사물인터넷 데이터 검색, 이종 시스템 데이터 통합 등 사물인터넷 데이터 활용과 관련 있는 분야의 기존 연구 분석을 바탕으로 도출한 것이다. 그리고 나머지 한 가지는 이 두 가지 기능을 바탕으로 데이터 분석을 수행하고자 할 때 필요한 일련의 작업을 자동화하는 기능이다.

이렇게 도출된 기능들의 개념과 설계 아이디어에 대해 본 장의 1절에서 개략적으로 기술하였고, 각 기능의 설계에 관한 연구내용들을 2-4절에 보다 구체적으로 기술하였다.

#### 1. 제안된 확장 기능

사물인터넷 서비스 플랫폼 설계 시 고려해야 하는 세 가지 기능은 첫째, 육하원칙 기반의 IoT 데이터 검색 기능, 둘째, 에이전트 기반 IoT 연합 DB 구성 기능, 셋째, IoT 데이터의 통계적 처리 기능이다.

##### 1) 육하원칙 기반 IoT 데이터 검색

민성현 등[37]에 따르면 다양한 도메인의 시스템이 상호작용할 수 있게 해야 한다는 점을 고려할 때, 서비스 플랫폼은 ‘이종 도메인 간 데이터 교환’이라는 유스케이스를 구현할 수 있어야 한다. 이는 ‘동일 도메인 시스템 간 데이터 교환’보

다 높은 수준의 상호운용성이 요구되는 작업이다. 도메인이 다르다는 것은 데이터를 취급하는 관점이나 관례, 용어 등 지식체계가 다르다는 것이며, 이 때문에 통신 프로토콜이나 CRUD (Create / Retrieve / Update / Delete) API 등을 제공하는 것만으로는 이종 도메인 간 원활한 데이터 교환을 기대하기 어렵다.

타 도메인의 데이터를 이용하고자 하는 경우 해당 도메인의 지식체계에 대한 학습이 필요할 것이며, 데이터 교환에 성공한 경우라도 대부분의 경우 데이터 변환과 같은 추가 작업을 거친 후에야 원하는 서비스에 적용 가능할 것이기 때문이다.

이러한 문제를 해결하기 위해서는 특정 도메인에 의존되지 않는 표준형식으로 데이터를 구조화할 필요가 있는데, 플랫폼이 보유하는 여러 데이터들 중에서도 실제 서비스 개발과 밀접한 관련이 있는 ‘측정데이터(센서에 의해 측정되는 데이터)’의 구조화 및 데이터 획득방식이 본 연구의 관심대상이다.

본 논문은 각 측정데이터들을 하나의 상황에 대한 설명으로 간주한다. 널리 알려진 바와 같이 완전한 상황 표현을 위해서는 육하원칙을 이용할 수 있다. 관련연구 분석을 통해 언급하였듯이 상황인지 데이터 모델 구축을 위해 육하원칙이 활용되고 있는 것을 확인할 수 있었다. 그러나, 사물인터넷 플랫폼의 데이터 모델은 아직 육하원칙에 대한 검색 기준을 충분히 제공하지 못한다는 점도 확인하였다. 이에, 본 논문에서는 데이터 사용자들이 시스템의 구성이나 해당 도메인의 지식체계에 대해 알지 못하더라도 육하원칙 기반으로 데이터를 검색할 수 있는 기능을 설계하였다.

이러한 기능은 데이터에 대한 접근성, 특히 검색의 용이성을 향상시키는 데 활용될 수 있으며, 결과적으로는 데이터 활용성 제고의 효과를 기대할 수 있다.

## 2) 에이전트 기반 IoT 연합 DB 구성

oneM2M[5], AllJoyn[6], IoTivity[7]와 같은 사물인터넷 서비스 플랫폼에 관한 표준기술들이 주로 다루고 있는 관심사는 앞으로 새롭게 구축할 사물인터넷 생태계에 대한 규격을 제시하는 데 있다.

그러나, 종래의 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[38], 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[39,40]이나, 필요한 데이터를 온전히 수집하기 어려운 상황에서의 해결책으로서 데이터 통합(Data fusion) 기법을 활용하는 방식[41] 등이 그것이다.

앞서 언급한 사물인터넷 표준화 기술들 역시 데이터의 공동 활용을 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.[42]

이는 데이터 품질 특성 중 유용성, 특히 충분성의 향상과 관련 있는 내용으로, 이러한 기능 활용 시 데이터 활용성 제고를 기대할 수 있다.

### 3) IoT 데이터의 통계적 처리

최근 빅데이터 기술이 다양한 분야에서 활용되고 있다. 앞서 언급한 두 가지 기능을 이용하여 구축한 사물인터넷 서비스 플랫폼은 다양한 분야에서 많은 양의 데이터를 수집, 관리하는 역할을 수행할 뿐 아니라, 데이터의 융합 활용이 가능할 것이기 때문에, 빅데이터 기술과의 업무 연관성을 충분히 고려하여 설계할 필요가 있다. 빅데이터 기술은 기본적으로 통계적 분석에 바탕을 둔 방법론이므로, 사물인터넷 플랫폼이 수집한 데이터를 통계적으로 처리할 수 있는 기능을 개발한다면, 융합 서비스 도출 시 보다 유용하게 사용될 수 있을 것이다.

서비스 플랫폼 계층에서 이러한 점이 고려된 기능을 제공할 경우 얻을 수 있는

효용은 기능적 유용성만이 아니다. 기존 방식에 따르자면, 원시 데이터가 확보되어 있는 경우, 그 형식의 통일성 여부와 상관없이 응용 계층에서 데이터 정제에 필요한 작업을 수행하면 된다. 그러나, 이 경우 서비스 개발자별로 같은 데이터에 대해서 비슷한 작업을 중복해서 수행해야 하고, 데이터의 용량이 큰 경우, 해당 작업에 소요되는 시간이 비즈니스 요구사항에 부적합해서 해당 서비스의 개발을 포기해야 하는 경우도 있을 것이다.

때문에, 서비스 플랫폼 계층에서 데이터 수집 시점, 또는 주기적 배치 작업 형식으로 기본적인 데이터 정제 작업을 수행해 놓는다면, 서비스 계층에서 보다 민첩하게 데이터를 활용할 수 있게 될 것이다.

본 논문에서는 특히, 데이터의 생성일시 속성에 초점을 맞추고 형식 통일을 구현하고자 한다. 또한, 데이터 간의 상관관계 분석에 주로 활용되는 통계 기법들을 염두에 두고 연구를 진행하였다.

이는 결과적으로, 데이터 과학에서 ‘탐험적 분석’이라고 부르는 작업에 필요한 기능을 사물인터넷 서비스 플랫폼 계층에서 제공할 수 있게 하는 방법에 관한 연구이며, 이 과정에서 데이터 시각화 기반의 시계열 분석과 데이터 분석가들이 가장 널리 활용하고 있는 통계 프로그래밍 언어인 ‘R[43]’을 활용하는 상관분석을 플랫폼 사용자가 별도의 시스템 없이 바로 수행할 수 있게 하는 것이다.

이러한 특징은 데이터 품질 특성 중 즉시성 향상과 관련된 것으로, 이 역시 데이터 활용성 제고의 효과를 기대할 수 있게 한다.

## 2. 육하원칙 기반 IoT 데이터 검색

사용자 친화적인 데이터 검색 기준을 제공하기 위하여 ‘육하원칙 기반 데이터 필터를 활용하는 확장된 IoT 자원 검색 모델’을 설계하였다. 현재의 oneM2M 표준을 기반으로 사물인터넷 플랫폼을 구성하고, 그 안에 다양한 도메인의 시스템들이 등록되어 데이터를 저장하고 있는 상황을 가정해보자.

데이터 분석가가 도메인이나 시스템에 관계없이 특정 지역에서 수집된 강우량과 하천 수위 데이터를 전부 조회하려고 한다면 먼저, 분석하고자 하는 데이터가 어느 도메인/시스템에서 관리되고 있는지 조사해야 한다. 다음으로 해당 도메인/시스템의 자원 관리 체계 및 용어체계를 각각 학습해야 하고, 각 시스템별로 조건을 설정하고 데이터를 선별하는 과정을 수행해야 한다. 또한 이러한 절차를 도메인/시스템의 수 만큼 반복적으로 수행해야 하며, <측정장소>와 같이 <Filter Criteria>로 제공되지 않는 조건요소의 경우 특별한 프로세스를 더 추가해야만 원하는 데이터를 얻을 수 있다.

그러나, 본 연구에서 제안하고자 하는 <Filter Criteria> 및 자원 관리 체계 확장안을 적용할 경우 사용자는 데이터 소스가 어느 도메인/시스템에 위치하는지, 해당 도메인/시스템의 지식체계나 시스템 구성 방식을 모르더라도 육하원칙 기반의 조건을 이용하여 한 번에 데이터를 가져올 수 있다.

### 1) 검색 기준 확장 요소 도출

사용자 친화적인 데이터 검색 기능 구현을 위해 <Filter Criteria>로 제공될 항목들을 추가한다. 이를 위해 육하원칙 질문 항목들을 총 여덟 가지 사물인터넷용 개념들에 사상한 결과는 Table 6과 같다. 이 때, 사물인터넷용 개념들이란 센서 데이터 수집을 주 목적으로 갖는 사물인터넷 기반의 시스템과 해당 도메인에서 공통적으로 취급하는 개념들을 가리킨다.

Table 6. 5W1H mapping results

5W1H	IoT Element	oneM2M Filter Criteria Expansion
When (existing)	measurement date and time	<b><i>createdBefore / createdAfter</i></b> - Existing entries use
Where (new)	measurement position (GPS)	<b><i>gpsLatitude, gpsLongitude, gpsRadius</i></b> - By entering the coordinates and the radius to support the search for the data measured at the radial position
	measurement position (zipcode)	<b><i>zipcodeAbove / zipcodeBelow</i></b> - By specifying the ZIP code coverage to retrieve measurement data to administrative district basis
Who (new)	measurement agents (management organizations of measurement system)	<b><i>owner</i></b> - Support to search by the management organization of the things - It should be managed as metadata - It should provide a list of available organizations as filter criteria
What (new)	measurement object (precipitation, temperature, etc.)	<b><i>measurementObject</i></b> - Support to search by specifying a measurement object - It should be managed as metadata - It should provide a list of available measurement objects as filter criteria
How (new)	measurement method (radar, pressure sensing, etc.)	<b><i>method</i></b> - Support to search by specifying a measurement method - It should be managed as metadata - It should provide a list of available measurement methods as filter criteria
	measurement equipment (equipment model name)	<b><i>equipment</i></b> - Support to search by specifying a measurement equipment - It should be managed as metadata - It should provide a list of available measurement equipments as filter criteria
Why (new)	measurement purpose (river monitoring, fire monitoring, etc.)	<b><i>purpose</i></b> - Support to search by specifying a measurement purpose - It should be managed as metadata - It should provide a list of available measurement purposes as filter criteria

## 2) 데이터 관리 모델

Table 6에서 제시한 Filter Criteria 항목 여덟 가지 중 일곱 가지는 기존에 없었으며, 본 연구에서 새로이 제안하는 것들이다. 따라서 새 조건 항목의 사용이 가능하도록 기존 자원 관리 모델도 함께 확장해야 할 필요가 있다. 이에 Fig. 15와 같이 확장된 자원관리모델을 제안하고 각 항목에 적용된 개념 및 고려사항들을 설명한다.

먼저, 측정주체, 측정대상, 측정방법, 측정장비, 측정목적 이 다섯 가지 조건요소들은 본 모델에서 메타데이터로서 취급된다. 메타데이터로 취급하는 이유와 방법에 대한 구체적인 설명을 위해 측정대상의 경우를 예시로 설명하면 다음과 같다.

각기 다른 제조사에서 생산된 강수량 관측용 장치가 특정 도메인 내에 공존하는 경우를 가정하자. 두 장치에서 생성되는 데이터의 유형을 플랫폼에서 각각 다른 이름으로 취급한다면(장치A: 강수량, 장치B: 강수량), 데이터 형식을 통일했다 하더라도, 데이터 유형을 나타내는 키워드 하나만으로는 동일한 데이터 유형을 갖는 두 데이터를 한 번에 조회할 수 없다. 또한, 데이터를 활용하고자 하는 사용자가 플랫폼 내에서 사용 가능한 데이터 유형이 무엇인지 설계도를 보기 전에 알 수 없을 수도 있다. 이런 경우를 방지하기 위해, 데이터 유형을 나타내는 데이터는 메타 데이터로서 관리하고, 플랫폼에 장치를 등록할 때마다 임의의 이름으로 데이터 유형을 등록하는 대신 메타 데이터로서 약속된 용어를 사용하여 등록하도록 구성한다.

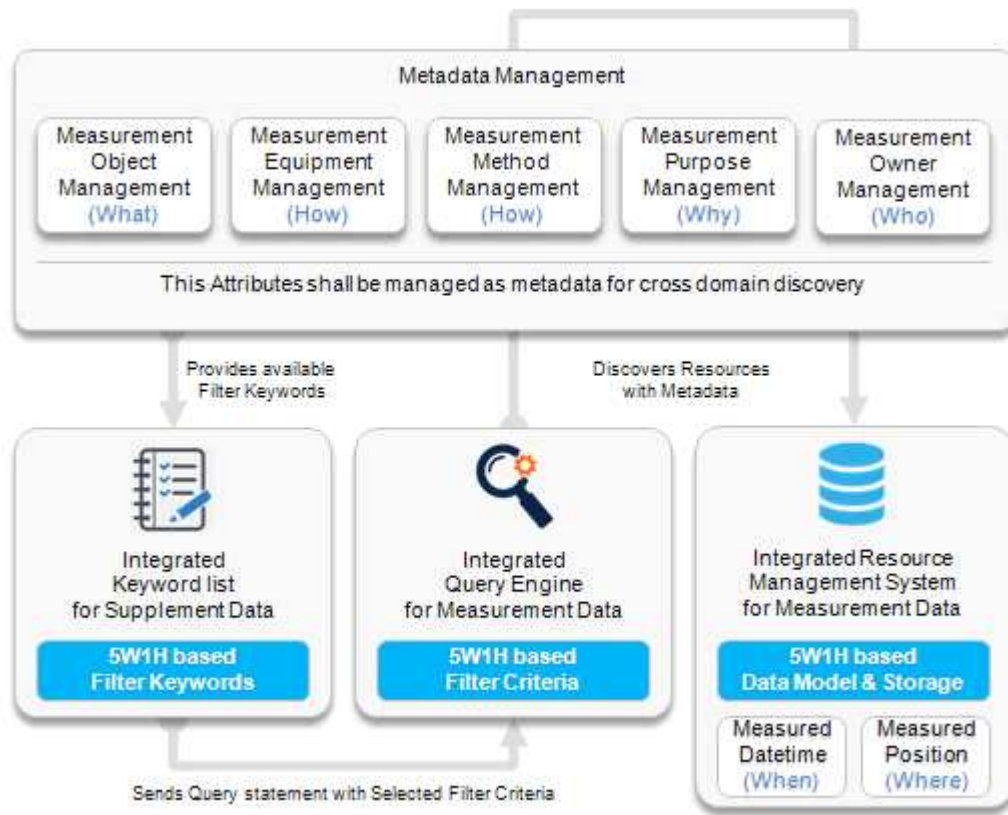


Fig. 15. Structure of resource management system

또한, 이렇게 관리되는 데이터 유형에 대한 명칭은 리스트 형태로 가용여부와 함께 표시되도록 하여 데이터를 이용하고자 하는 이가 이를 Filter Criteria로서 사용할 수 있게 구성한다.

Filter Criteria가 선정되고 쿼리문장이 입력되면 쿼리엔진은 메타데이터 관리시스템에서 데이터 조회에 필요한 키(데이터 ID 등)를 찾아 자원관리시스템에서 해당 데이터를 찾아낸다.

이 때, 측정시각 및 측정위치에 대한 정보는 메타데이터 관리시스템이 아닌, 자원 관리 시스템 내에 측정데이터와 함께 <contentInstance>의 속성으로서 관리되게 구성하는 것이 중요하다. 측정시각 속성은 이미 oneM2M 표준에서도 이러한 형태로 관리하도록 구성되어 있으나, 측정위치 데이터는 별도로 관리되도록 정의되어 있는데, 이는 하나의 사물에 다수의 장치가 연결되어 있더라도 위치 데이터는 공통으로 사용할 수 있으므로, 위치 정보를 공유할 수 있는 형태로 정의



한 것으로 볼 수 있다. 그러나, 고정형이 아닌 이동형 장치의 경우 시시각각 위치정보가 갱신되므로, 검색 메커니즘 관점에서 이 경우의 위치정보는 시간정보와 마찬가지로 특성을 갖는데 ‘위치정보 일치여부를 먼저 검사할 경우 그 결과값의 수가 최종 검색될 데이터의 수와 같거나 약간 적은 정도가 된다는 것이다. 이 경우 실제 데이터를 찾기 위한 쿼리가 첫 번째 쿼리를 통해 얻은 위치정보 결과들의 수만큼 많아지게 되므로, 검색 효율을 고려할 때, 데이터 중복을 감수하더라도 반정규화(denormalization)를 수행하여 위치정보가 측정데이터와 같은 자료구조(테이블 등) 내에서 관리되도록 해야 한다.

그리고, 본 자원 관리 모델에서 모든 측정데이터는 원자화된 상태로 관리해야 한다. 이는 다양한 조건으로 데이터를 조회할 수 있게 하고, 나아가 다차원적으로 데이터를 재조합하여 분석하거나 맥락을 파악하는 데 활용할 수 있게 한다. 하나의 사물이 보고하는 데이터 내에도 여러 가지 센서의 측정데이터가 복합적으로 포함되어 있을 수 있지만, 이들을 완전히 원자화하여 물리적 연결관계와 상관없이 각각의 데이터 소스를 독립된 것으로 취급할 수 있도록 논리 레벨에서 가상화해야 한다.

### 3) 육하원칙 기반의 데이터 검색 기법

Fig. 16은 앞서 설명한 자원관리모델의 각 구성요소들을 통해 육하원칙 검색이 수행되는 절차를 표현한다.

먼저, 시간 및 장소를 기반으로 데이터를 검색하고자 하는 경우 본 연구에서 제안하는 모델은 자원 저장소에 측정 데이터를 측정시각 및 장소정보(gpsPosition, zipCode)와 함께 저장하므로, 쿼리엔진이 다른 구성요소에 대한 참조 없이 저장소에 바로 질의를 수행한다. 이는 oneM2M의 기존 자원관리개념을 확장하여 위치정보를 중복 저장하는 방식으로 반정규화를 적용함으로써, 쿼리 수행 절차를 단축하는 효과를 얻을 수 있게 한 것이다.

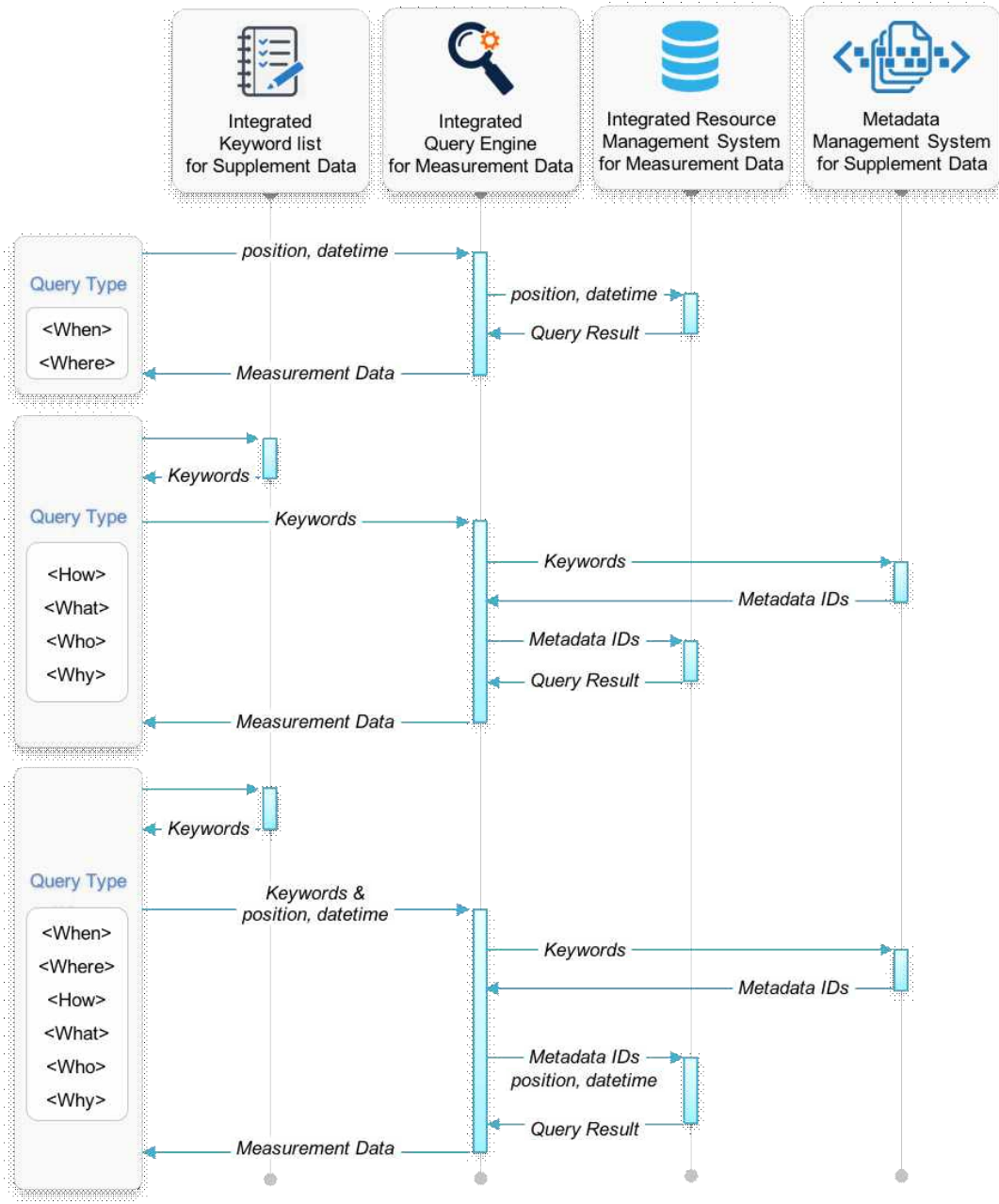


Fig. 16. Data search processes based on 5W1H

다음으로, 측정주체, 측정방법, 측정장비, 측정대상, 측정목적 등 측정데이터에 대한 보충 자료를 이용하여 데이터를 검색하는 경우, 사용자는 검색조건으로 사용할 수 있는 키워드 리스트를 시스템으로부터 제공받아 원하는 조건에 해당하는 키워드 조합을 작성하여 쿼리 엔진에 전달한다. 쿼리엔진은 이 키워드 조합을

이용하여, 메타데이터 관리시스템을 통해 해당 키워드에 해당하는 메타데이터들의 식별자를 가져와 측정데이터 검색에 이용할 쿼리문을 작성하고, 자원저장소를 대상으로 최종 쿼리를 수행한다.

마지막으로, 육하원칙의 모든 질의요소를 사용하는 쿼리의 경우 전술한 두 유형의 순서를 결합한 형태의 절차가 구성되며, 시각 및 위치정보는 측정데이터와 함께 저장되어 있으므로, 메타데이터로 취급되는 다섯 가지(측정대상, 측정주체, 측정방법, 측정장비, 측정목적)를 검색조건으로 사용하기 위해 식별자를 얻어오는 절차가 먼저 수행되고, 최종 쿼리 실행 시 이 식별자들과 위치, 장소 조건을 결합한 쿼리가 활용된다.

### 3. 에이전트 기반 IoT 연합 DB 구성

각기 다른 구조와 운영환경을 갖고 있는 다양한 이종 시스템들을 제한적으로나마 신규 시스템과 연동하여 통합된 사물인터넷 서비스 플랫폼으로 구축할 수 있게 하는 방법에 관해 연구하였다. 이 때, 통합 대상이 되는 이종 시스템은 RDBMS를 이용하여 데이터를 관리하고 있는 경우를 가정하여 설계를 진행하였으며, 결과적으로는 다수의 통합 대상 이종 시스템들의 RDBMS 내에 저장된 데이터들을 해당 시스템의 구조적 특성과 무관하게 통합하여 활용할 수 있는 모델을 제시하였다.

이를 위해, 연합 데이터베이스 시스템(Federated Database System)[11] 구성개념을 활용하였다. 이종 시스템들이 갖는 구조적 다양성을 수용할 수 있도록 데이터 조회 인터페이스를 추상화함으로써, 레거시 시스템의 데이터 소스가 신규 플랫폼 내에 존재하는 것처럼 가상 통합을 이루는 구조를 설계하였다. 이 과정에서 CUAHSI ODM[26]이 제시하는 데이터 모델 설계 개념을 참조하였다. 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[31]의 설계 개념 또한 참조하였다.

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.

## 1) 제안된 연합 DB 구성 방안

본 연구에서 제시한 이중 시스템의 IoT 데이터 연합 및 통합 질의 시스템의 설계 개념은 Fig. 17과 같다. 이 시스템은 하나의 Data Federation Management System(이하 DFMS)과 다수의 Data Federation Agent(이하 DFA)의 연결로 구성된다.

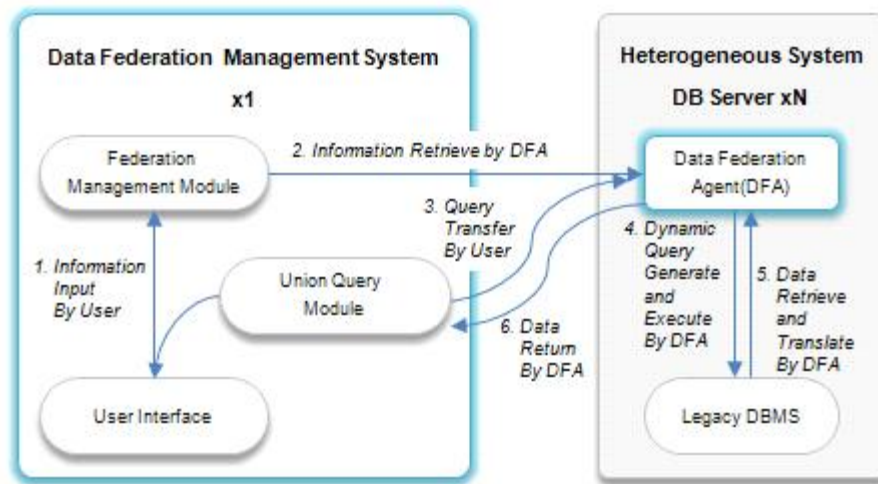


Fig. 17. IoT data federation concept

## 2) 데이터 연합 관리 시스템

DFMS(Data Federation Management System)는 여러 시스템에 흩어져 있는 데이터에 대한 연합 구성을 가능하게 하며, DFMS의 핵심 구성요소는 연합관리 모듈과 통합질의모듈이다. 연합관리모듈은 연합 대상인 이중 시스템들을 연결하고 데이터를 검색할 때 필요한 정보들을 관리하며, 통합질의모듈은 사용자로부터 입력된 질의를 분석한 후, 해당 데이터를 보유하고 있는 시스템들을 찾아서 질의 조건을 전달한다. 또한, 질의를 실행한 시스템들로부터 질의 결과를 반환 받아 병합하여 사용자에게 전달하는 작업도 수행한다.

### 3) 데이터 연합 에이전트

DFA(Data Federation Agent)는 통합 대상 이종 시스템의 DB서버에 설치되는 에이전트 프로그램으로서, DFMS로부터 해당 이종 시스템의 정보를 전달받아 해당 시스템에서 동작할 수 있는 형태의 쿼리문을 동적으로 생성하고 레거시 DB의 데이터를 표준 데이터 모델용으로 변환 후 반환하는 기능을 수행한다.

이 시스템에서의 데이터 흐름은 연합하고자 하는 이종 시스템에 대한 정보 입력으로부터 시작된다. 이렇게 입력된 정보는 DFMS에서 관리되는데, 추후 DFA가 설치된 시스템이 구동되면 DFA의 요청에 의해 해당 DFA에 전달되어 자동으로 연합설정이 구성된다. 이 상태에서 사용자가 데이터 조회 명령을 입력하면 이 명령이 통합 질의 모듈에 의해 해당 데이터를 보유한 시스템의 DFA로 전달되고, DFA는 해당 시스템의 환경정보를 가지고 동적으로 쿼리를 구성하여 실행한 후 조회된 데이터를 통합 질의 모듈을 통해 사용자에게 반환한다.

### 4) IoT 연합 DB 기반 통합 질의

사용자는 찾고자 하는 데이터가 어떤 시스템에 어떤 형태로 존재하고 있는지 모르더라도 측정유형, 측정일시, 측정장소를 조건으로 하여 질의를 작성할 수 있는데, 이처럼 실제로는 분리되어 있는 시스템들을 논리적으로 통합된 것처럼 활용할 수 있게 하는 방법을 본 연구에서는 ‘실시간 가상 통합’라 하며, 앞서 기술한 데이터 연합 모델에 의해 실시간 가상 통합이 구현된다.

또한, 실시간 가상 통합 환경 하에서 Table 7의 예시와 같이 데이터를 보유한 이종 시스템의 종류나 데이터베이스의 구조에 대해서는 신경 쓰지 않고도 필요한 조건만으로 질의하게 하는 방법을 ‘통합 질의’라 한다.

이러한 통합 질의를 가능하게 하는 구성요소인 DFMS와 DFA의 구현 및 실험 결과에 대해서는 이어지는 장들을 통해서 보다 상세하게 설명한다.

Table 7. 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)

#### 4. IoT 데이터의 통계적 처리

##### 1) 제안된 통계 처리용 데이터 모델

원시데이터를 통계 분석에 활용하기 위해서는 데이터 속성을 통일하는 과정이 필요하다. 본 연구에서는 ‘육하원칙 기반 IoT 데이터 검색 기능’의 개념 및 설계를 설명하면서 이미 데이터 통합용 형식을 제시하였으나, 이는 스키마 및 데이터 단위 형식의 통일에 관한 것이었다.

데이터들을 통계적 분석에 활용하고자 할 때는 데이터 기록 일시(Datetime)가 중요한 요소로 활용되며, 특히 서로 다른 소스로부터 생성된 데이터들 간의 상관관계를 조사하고자 할 때는 양 측 데이터가 동일한 기록일시 속성을 갖고 있어야 용이하게 분석할 수 있다. 이 절에서 설명하고자 하는 데이터 속성 통일이 바로 생성 일시를 맞추는 작업에 관한 것이다.

데이터들은 그 생성주체나 생성 목적에 따라서 각기 다른 시간주기로 생성되며, 어떤 경우에는 정해진 주기 없이 수시로 생성되기도 한다. 이런 경우 각기 다른 시점에 생성된 데이터들로는 상관관계를 조사하기가 어려운 문제가 발생한다.

예를 들어, 주차 감지 데이터는 일정한 주기 없이 차량이 감지될 때마다 생성되고, 기온 데이터는 매 10분 주기마다 생성되는 상황이라고 가정하자. 이러한 상황에서 분석가가 주차와 기온 간의 상관관계를 조사하고자 한다면, 단순히 원시데이터만으로는 비교의 기준을 찾을 수 없기 때문에 어떤 식으로든 기준을 찾아 맞추는 추가 작업을 수행할 것이다.

본 연구를 통해 수행한 내용은 이러한 시점 기준 통일에 관한 것으로서, 데이터가 수집되는 시점에 여러 가지 기초 통계를 생성하도록 하여, 분석가는 추가 작업 없이도, 미리 준비된 다양한 기준을 이용할 수 있게 하였다.



## 2) 시간대별 기초통계 자동 사상

제시하는 아이디어는 기간 단위별로 기초통계를 미리 생성해 뒀다가, 비교 시에 활용하는 방법이며, 1분부터 1년까지 Table 8과 같이 기간 기준을 정하고, 해당 시간기준별로 다섯 가지 통계 데이터를 생성해두는 것이다.

Table 8. Basic statistics definitions

Div.	Definitions
Statistical Period	1minute, 10minutes, 1hour, 1day, 1month, 1year
Type of Statistic	Maximum, Minimum, Average, Sum(value), Sum(Quantity)

Table 8의 기간 기준에 대해 설명하자면, 모든 사물의 데이터는 각 사물이 생성한 데이터를 수집하여 저장하면서 매 1분마다 해당 1분 동안 수집된 데이터에 대해 최대값, 최소값, 평균값, 값 합계, 생성건수 합계를 산출하여 통계용으로 준비된 테이블에 저장한다. Fig. 18은 이러한 통계용 데이터를 관리하는 테이블의 구조를 나타낸다. 이 테이블은 모든 센서 데이터를 대상으로 각 센서 데이터에 대한 특정 시점에서의 다섯 가지 기초 통계 산출 값들을 관리할 수 있도록 설계되었다.



Fig. 18. Basic statistics management schema

### 3) 기초통계 활용 프로세스

Fig. 19는 시간대별 기초통계 자동 생성 및 그 결과 데이터의 활용 개념을 도식화한 것이다. 원시데이터가 저장되는 시점에 원시데이터 및 해당 데이터에 대한 기초 통계 데이터도 함께 저장되고, 데이터 요청이 있을 시에는 원시데이터뿐만 아니라 기초통계 데이터까지도 함께 검색하여 반환할 수 있는 구조를 갖는다. 이러한 구조를 통해 사용자는 원시 데이터만으로 비교분석이 어려운 경우에도, 통계데이터를 이용하여 분석을 수행할 수 있게 된다.

그리고, 이 과정에서 반환되는 데이터는 사용자가 원하는 형식으로 변환 제공이 가능해야 한다. 본 연구에서는 통계 분석용으로 즉시 사용할 수 있게 하기 위해 관계형 데이터베이스에서 관리하는 기본 형식인 튜플 형태가 아닌, 매트릭스 형태의 데이터 세트 생성을 지원하도록 설계하였다.

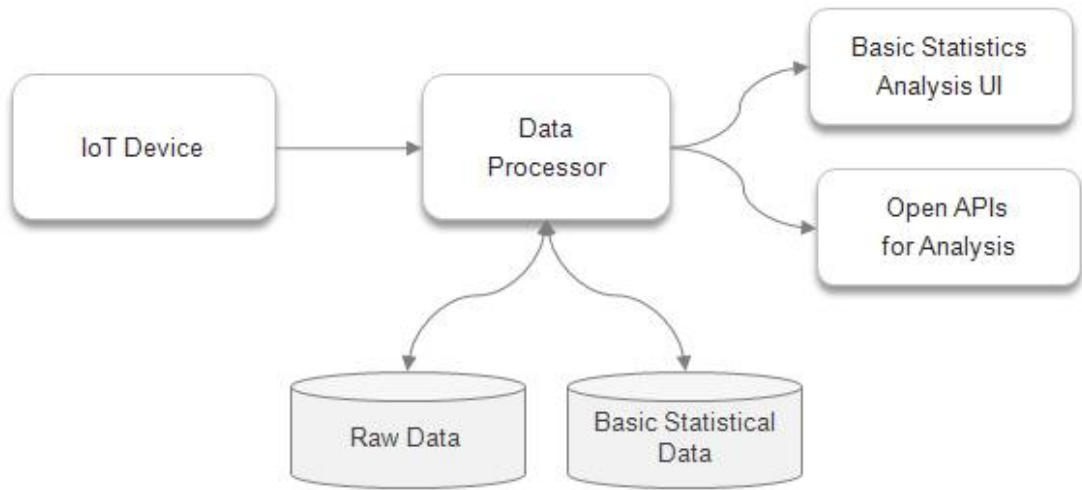


Fig. 19. Data store/retrieve structure for statistic analysis

## IV. 확장 기능 구현

본 연구에서 설계한 내용은 자체 개발한 사물인터넷 서비스 플랫폼인 ‘만다린 플랫폼’을 기반으로 기능을 확장하는 방식으로 구현하였고, 각 기능은 이 플랫폼 상에서 완전하게 동작할 수 있는 상태로 구현하였다.

### 1. 기능 구현 환경

본 연구를 위해 분산컴퓨팅 환경을 지원하는 사물인터넷 서비스 플랫폼인 ‘만다린 플랫폼’을 기반구조로 활용하였으며, 이 플랫폼의 주요 기능에 대하여 다음과 같이 기술하였다. Fig. 20은 만다린 플랫폼의 대시보드로서, 플랫폼에 연결된 자원들의 현황을 한 눈에 조회할 수 있게 한다.

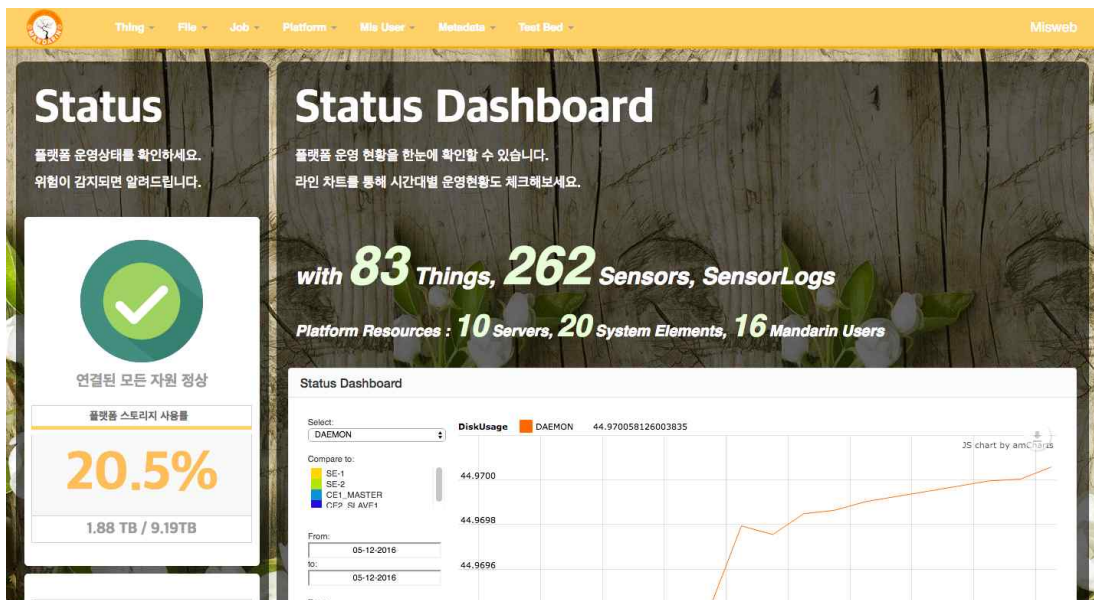


Fig. 20. Dashboard of mandarin platform

대시보드에서 확인할 수 있는 내용들은 플랫폼에 연결된 Thing과 Sensor의 수, Sensor로부터 수집된 SensorLogs의 수, Server 및 User의 수 등이다. 또한, 연결된 구성요소들의 가용상태, 사용률 등을 시각화해서 표시해준다.

만다린 플랫폼은 사물인터넷 세계를 Thing과 Sensor들 간의 관계, 또는 Actuator들 간의 관계로 정의하도록 설계되었으며, Fig. 21 - 24에서 보이는 것처럼 Things & Sensors 현황, Things, Sensors, SensorLogs를 관리할 수 있다. 만다린 플랫폼은 이처럼 기본적인 사물인터넷 서비스 플랫폼으로서의 기능을 갖추고 있을 뿐만 아니라, 분산 컴퓨팅 환경 하에서 사물들의 데이터를 효과적으로 취급할 수 있게 하는 기술도 제공하고 있다.

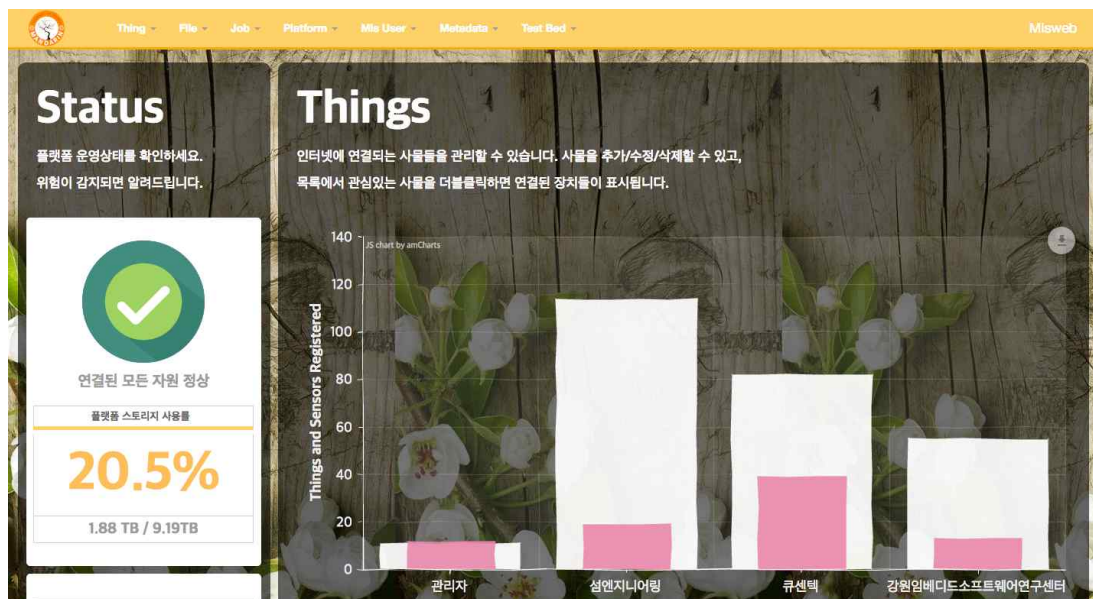


Fig. 21. Things & sensors status UI of mandarin platform

	Id	Thing Name	Owner	Thing Type	Zip Code	Is Moveable			
<input type="checkbox"/>	131	Aqua 수조	gemscrc	100		false			
<input type="checkbox"/>	130	AWS-885	gemscrc	AWS	33.3315	false			
<input type="checkbox"/>	129	AWS-871	gemscrc	AWS	33.3624	false			
<input checked="" type="checkbox"/>	128	AWS-870	gemscrc	AWS	33.3698	false			
<input type="checkbox"/>	127	AWS-855	gemscrc	AWS	33.1664	false			
<input type="checkbox"/>	126	AWS-793	gemscrc	AWS	33.2167	false			
<input type="checkbox"/>	125	AWS-792	gemscrc	AWS	33.3536	false			
<input type="checkbox"/>	124	AWS-782	sumeng	AWS	33.3852	false			
<input type="checkbox"/>	123	AWS-781	sumeng	AWS	33.5199	false			
<input type="checkbox"/>	122	AWS-780	sumeng	AWS	33.28	false			

Fig. 22. Things management UI of mandarin platform

# Sensors

사물에 연결된 센서들을 관리하고, 센서 데이터를 조회할 수 있습니다.  
목록에서 관심있는 센서를 더블클릭하면 센서 데이터가 표시됩니다.

**Sensors**

10 records per page Search:

	Id	Name	Sensor Type	Thing	Sensor Id	Measured Value			
<input type="checkbox"/>	294	water	1001	Aqua 수조_131	aqua01				
<input checked="" type="checkbox"/>	293	DO	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	292	ph	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	291	temp	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	290	food	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	289	salinity	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	288	nitrate	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	287	HNO2	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	286	NH3	1001	Aqua 수조_131	aqua01				
<input type="checkbox"/>	285	aws-130-43	aws-windspeed	AWS-885_130	885				

Fig. 23. Sensors management UI of mandarin platform

Id	Measured Datetime	Measurement Type	Measured Value	Latitude	Longitude	Thing Id
22661085	2016-11-29 21:43:47	EC(전기전도도)	1.69	33.4318562	126.92552249999994	98
22661084	2016-11-29 21:43:47	weight	694	33.4318562	126.92552249999994	98
22661083	2016-11-29 21:43:47	weight	5303	33.4318562	126.92552249999994	98
22661082	2016-11-27 17:02:47	position	11	33.4318562	126.92552249999994	32
22661081	2016-11-27 17:01:32	position	12	33.4318562	126.92552249999994	32
22661080	2016-11-27 16:59:06	position	11	33.4318562	126.92552249999994	32
22661079	2016-11-27 16:59:06	urgent	0	33.4318562	126.92552249999994	32
22661078	2016-11-27 16:58:37	position	12	33.4318562	126.92552249999994	32
22661077	2016-11-27 16:58:36	position	12	33.4318562	126.92552249999994	32
22661076	2016-11-27 16:58:35	urgent	0	33.4318562	126.92552249999994	32

Fig. 24. Sensor logs management UI of mandarin platform



## 2. 육하원칙 기반 IoT 데이터 검색 기능

사용자 친화적 데이터 검색 기능 구현을 위해 III장에서 설계한 육하원칙 기준들 중 가장 핵심적인 데이터 정의 요소인 측정일시, 측정위치, 측정유형에 대한 데이터 관리용 스키마를 Fig. 25와 같이 구현하였다. 여기서 latitude, longitude, zip\_code는 측정위치를, measurement\_type은 측정유형을 나타내는 요소들인데, 이 정보들은 정규화 과정을 통해 별도의 스키마로 분리할 수도 있지만, 데이터 조회 성능 향상을 위해 반정규화 원칙을 적용하여 측정된 데이터와 동일한 스키마 내에서 관리되도록 구현하였다. 또한 이 외의 육하원칙 요소들은 별도의 스키마를 통해 메타데이터로서 관리되도록 구현하였다.

Field	Type	Length
id	BIGINT	20
measured_datetime	DATETIME	
latitude	DOUBLE	
longitude	DOUBLE	
zip_code	INT	11
measured_value	VARCHAR	255
measurement_type	BIGINT	20
sensor	BIGINT	20

Fig. 25. Table schema for sensor data

이 때, 측정위치에 대한 정보는 GPS 좌표와 우편번호를 선택적으로 이용할 수 있도록 하였는데, 데이터 검색 시 GPS 좌표는 특정 지점의 좌표와 반경을 지정하는 방식으로 사용할 수 있고, 우편번호는 특정 행정구역 단위로 데이터를 필터링하기 위해 사용할 수 있다.

이러한 위치정보는 Fig. 26과 같이 해당 데이터를 생성한 센서를 소유하고 있는 사물(Thing) 정보에 포함된 위치정보로부터 복사되어 데이터 저장 시 함께 저장되도록 구현하였다.



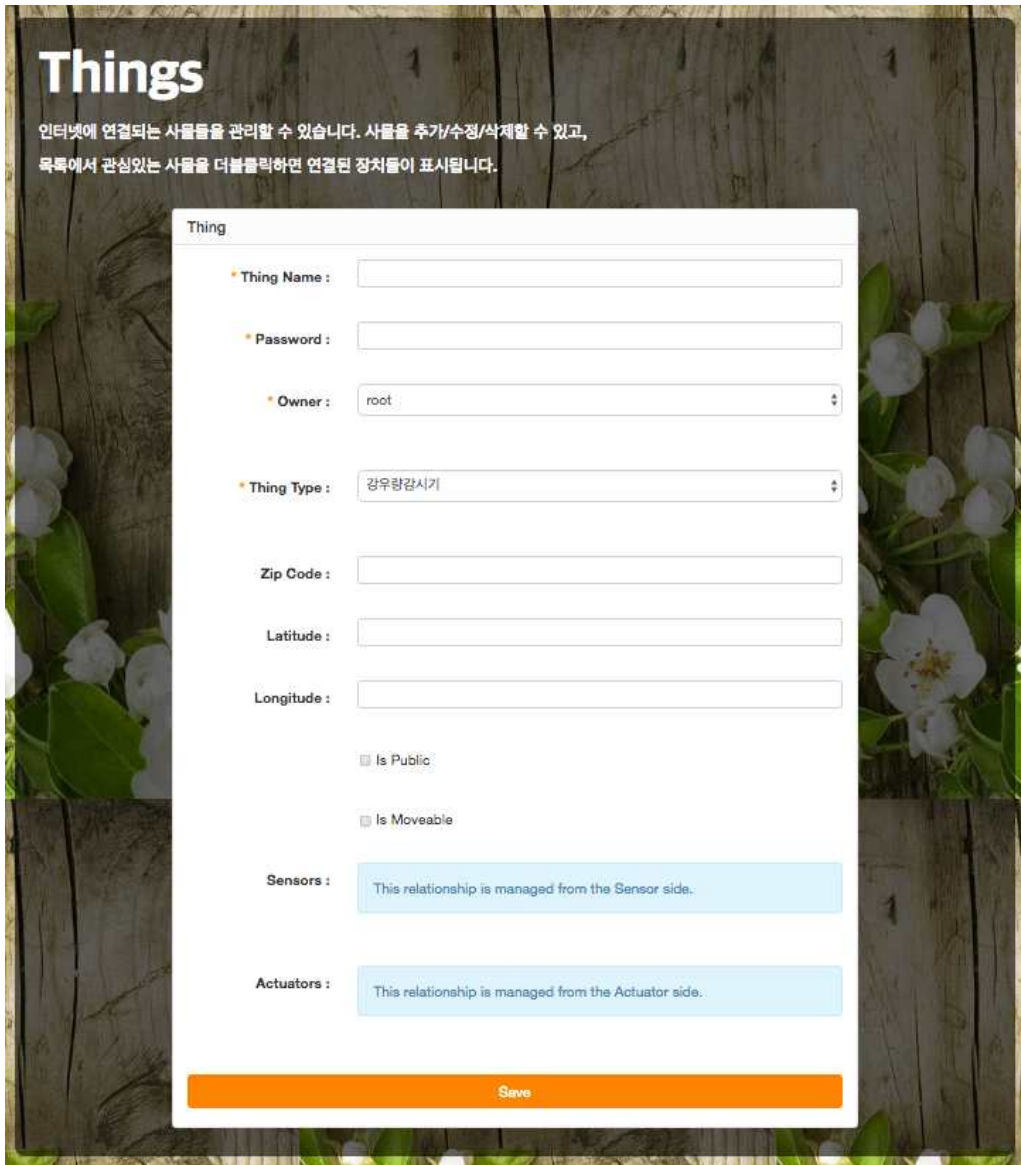


Fig. 26. Thing registration UI

또한, 측정유형 정보의 경우 Fig. 27과 같이 메타데이터로서 관리되도록 하며, 이를 센서 정보에 사상하여 관리할 수 있게 구현하였다. 이런 과정을 통해 사용자는 측정위치, 측정유형, 측정일시와 같이 사용자 친화적인 필터규칙을 이용하여 데이터 필터링을 수행할 수 있게 하였다. Fig. 28은 이런 방식의 데이터 필터링 기능을 제공하는 사용자 인터페이스이다.

# Measurement Type

센서가 취급하는 측정의 종류를 관리할 수 있습니다.

수집된 데이터를 조회할 때 이 여기에 입력된 측정 종류를 활용할 수 있습니다.

### Measurement Types

#### Measurement Type

\* Measurement Type Id      \* Measurement Unit      Numerical

Search:

<input type="checkbox"/>	Id	Measurement Type Id	Measurement Unit	Numerical	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	47	aws-windspeed	m/s	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	46	aws-winddirection	degree	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	45	aws-humidity	%	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	44	aws-temperature	C	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	43	aws-rainfall-yn	yn	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	42	aws-rainfall	mm	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	41	rain_YN	1/0	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	40	pressure	hPa	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	39	humidity	%	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>
<input type="checkbox"/>	38	winddirect	degree	true	<input type="checkbox"/>	<input type="button" value="edit"/>	<input type="button" value="delete"/>

Fig. 27. Measurement type management UI

Find Sensor Logs

\* Measurement Type :   
waterlevel  
waterspeed  
rainfall  
snowfall

\* Start Date(ex: 5/1/16) :

\* End Date(ex: 5/30/16) :

\* Start ZipCode (ex: 60001) :

\* End ZipCode (ex: 69999) :

Fig. 28. User friendly filter UI

### 3. 에이전트 기반 IoT 연합 DB 구성 기능

#### 1) 데이터 연합 관리 시스템

DFMS는 사용자 인터페이스, 연합 관리 모듈, 통합 질의 모듈을 포함하는 웹 기반 애플리케이션으로 구현하였으며, 구축하고자 하는 사물인터넷 서비스 플랫폼 내에 설치되어 동작하도록 하였다. DFMS 개발 및 실행환경은 Table 9와 같다. DFMS는 이중 시스템 연동에 필요한 정보들을 관리하는 기능을 필요로 하는데, 이러한 기능들은 ‘연합 관리 모듈’로서 구현하였다.

Table 9. Development and experiment environment of DFMS

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 10과 같다. Table 10에 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.

본 논문에서는 구현하지 않았지만, Table 10에 표현한 정보 외에, Who, What 에 대한 정보도 추가적으로 관리가 가능하다. 전통적인 수직 구조 서비스 플랫폼 에서는 그 운용 주체나 운용 목적에 대해 별도로 관리할 필요가 없었기 때문에,

레거시 시스템들의 데이터베이스에는 해당 정보들이 존재하지 않는 경우가 대부분이지만, 이러한 정보들도 수평 구조의 플랫폼으로 통합할 때에는 유용하게 활용될 수 있으며, 데이터 연합 구성 시점에 메타데이터로서 입력하여 추후 검색 기준으로 활용할 수 있게 할 필요가 있다. 이는 데이터 분석 과정에서 기존에 발견하지 못했던 새로운 정보로 가공될 가능성을 내재하고 있기 때문이다.

Table 10. Heterogeneous system's information for federation

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 Schema	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.
DB Access Info.	unitTransformCoefficient	a multiply Coefficient that will needed value standardization ex: cm → m : x0.01, kg → g : x0.001
	databaseUrl	DB access path
	databaseUsername	username for DB access
	databasePassword	password for DB access

이 정보들을 입력하고 나면 이중 시스템의 레거시 DB 서버에 DFA를 설치할 수 있다. 이때, 레거시 시스템에서의 작업 요소를 최소화하기 위해 설정과정도 외부에서 수행할 수 있게 구성하였는데 ‘연합 관리 모듈’이 관리하는 Table 10의

정보들이 DFA에도 전달된다. DFA는 매 실행시점마다 DFMS에 접근하여 자신이 접근해야 할 DB에 대한 접속정보 및 DB 스키마 정보를 요청 및 확인한 후 데이터 접근에 필요한 준비 작업을 수행하는 방식이다. DFMS는 정보를 요청하는 DFA의 macAddress를 이용하여 해당정보를 찾아 전달하도록 구현하였다.

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

DFMS의 또 다른 구성요소로 ‘통합 질의 모듈’이 있다. 이 모듈은 사용자가 데이터 조회 명령을 입력했을 때 이중 시스템들을 가상으로 통합하여 일괄적으로 조회를 수행하도록 구현하였다. ‘통합 질의 모듈’은 ‘연합 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.

이 과정에서 질의와 응답에 소요되는 시간을 단축시키기 위해 파이프라인 구조를 적용하였다. 다수의 이중 시스템에 대해 질의를 순차적으로 수행하는 경우 불필요한 시간 지연이 발생할 수 있으므로, 데이터를 보유한 이중 시스템들에게 일괄적으로 쿼리를 전송하고, 반환되는 데이터들을 임시저장소에서 병합한 후 사용자에게 제공하는 방식이다.

이와 같이 전술한 구성요소들의 기능을 활용하면 사용자는 원하는 데이터의 속성만을 정의하는 방식으로 통합 질의를 실행할 수 있다. 여기서 데이터의 속성이란 관측유형, 관측시점, 관측위치를 의미한다. 이 시스템은 ‘연합 관리 모듈’과 ‘통합 질의 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 질의 결과로 반환받을 수 있게 된다.

Fig. 29는 통합하고자 하는 이중 시스템의 데이터베이스에 대한 연동설정 화면의 일부분이다. 이 화면을 통해 입력하는 데이터 중 Table Name, Entity Id

Field Name, Sensor Id Field Name, Value Field Name, Dt Field Name은 동적 쿼리문 생성용으로 사용될 정보이다.

그리고, Dt Format Pattern은 시간형식 통일을 위해 사용하는 정보이다. 플랫폼에서 기본으로 사용하는 시간형식과 이종 시스템 시간 형식이 상이하더라도 이 때 입력되는 시간형식 패턴을 참조하여 표준형식으로 변환 후 처리하도록 구현하였다. 이 때, 사용자는 자바 언어의 시간 표현형식을 이용하여 연동하고자 하는 이종시스템의 DB에서 정의하고 있는 시간의 형식을 기술할 수 있다.

**IoT-SEs**  
IoT-SE : 레거시 시스템의 사물들도 연결합니다.  
목록에서 관심있는 사물을 더블클릭하면 연결된 장치들이 표시됩니다.

IoT Se

Ip Address :

Mac Address :

Server : DAEMON

Is Enabled

Name :

Sync Interval : 0

Owner : root

Table Name :

Entity Id Field Name :

Sensor Id Field Name :

Value Field Name :

Dt Field Name :

Dt Format Pattern :

Dialect : MySQLInnoDBDialect

Db Url :

Fig. 29. Database integration setting UI for legacy system DB



Unit Transform Coefficient는 단위 변환을 위한 계수이다. 데이터의 유형이 수치형일 경우, 플랫폼에서 기본으로 취급하는 단위와 레거시 시스템에서 취급하는 단위가 상이할 경우, 예를 들면 mm와 Cm와 같이 상이한 단위를 사용하여 데이터를 표현하는 경우를 말한다. 이 경우 Unit Transform Coefficient로 설정된 수치를 해당 데이터에 곱하는 방식으로 단위 변환을 수행한다. Cm를 mm로 바꾸기 위해서 사용할 변환계수는 10이 되는 식이다.

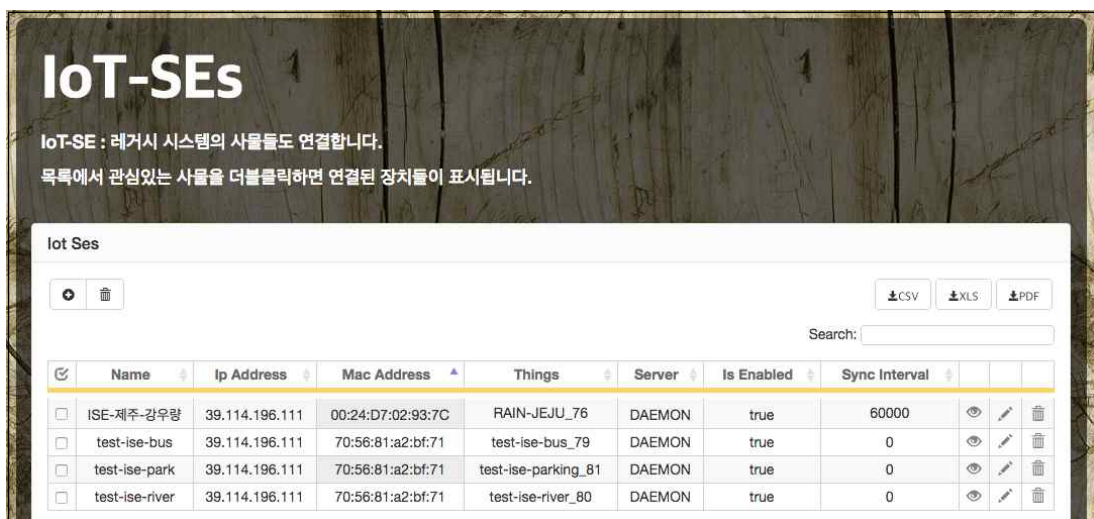


Fig. 30. Legacy heterogeneous systems federation management UI

그 밖의 정보들은 DB 접속에 필요한 정보들인데, 그 중 Dialect는 DBMS 종류를 표현하는 정보로서, 여기에 설정한 정보대로 동적으로 생성되는 쿼리문의 형식도 바뀌게 되는 구조이다. Fig. 30은 등록된 레거시 시스템들의 정보이다.

## 2) 데이터 연합 에이전트

DFMS와 이종 시스템의 연동을 위해 레거시 시스템 내에 설치하는 프로그램인 DFA를 추가로 구현하였으며, Fig. 31은 DFA의 내부 구성요소들을 표현한다. DFA는 이종 시스템의 레거시 DBMS에 직접 접근하는 프로그램으로서, DFA를 구현함에 있어서 가장 중요한 고려 요건은 이종 시스템의 운영환경에 대한 수정



은 최소화하면서 설치 및 실행이 가능해야 한다는 점이였다. 또한, 레거시 시스템들의 운영환경이 모두 상이하므로 DFA의 소스코드를 수정하지 않고 쉽게 적용할 수 있도록 범용성을 강화하는 방안도 고려하였다. 이를 위해 DFA는 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 DFA 자체의 구동환경을 설정하는 일도 해당 이종 시스템 내에서 수행되지 않도록 DFMS에 저장된 정보를 DFA의 Connection Configurator가 읽어오는 방식으로 구현하였다.

DFA는 DFMS의 ‘통합 질의 모듈’로부터 수신되는 질의 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.

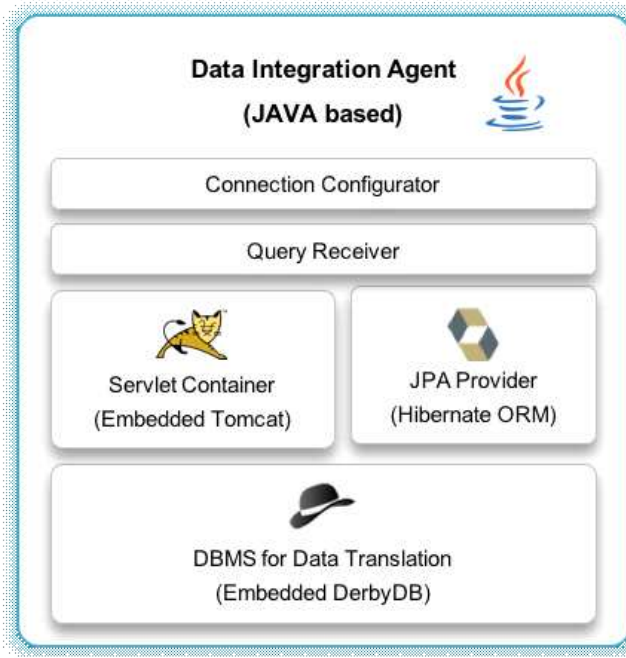


Fig. 31. Structure of data federation agent

웹 API를 통해 수신된 질의 요청은 Java Persistence API[44] Provider인 Hibernate[45]를 통해 레거시 DBMS에 맞는 쿼리문으로 자동 변환 및 실행되도록 구현하였다. Hibernate는 현존하는 대부분의 주요 RDBMS에 대한 맞춤형 쿼리문 생성을 지원할 수 있다. Hibernate의 이러한 기능을 이용하여 Connection Configurator가 DFMS로부터 읽어 들인 레거시 DBMS의 유형 및 스키마 정보를 이용하여 레거시 데이터를 본 시스템에서 미리 정의해 놓은 데이터 스키마에 맞게 변환하고 결과로 반환하는 기능을 구현했다. 이 때, 데이터 변환은 시간형식 통일과 값 단위(예: kg → g, mm → m 등) 통일 작업을 포함함으로써, 사용자에 의한 데이터 정제 과정을 최소화할 수 있게 구현하였다.

또한, 외부 요청에 빠르게 응답할 수 있도록 필요시 레거시 데이터를 미리 변환하여 내장DB인 DerbyDB[46]에 저장할 수 있도록 구현하였다. DerbyDB는 Embedded Tomcat과 마찬가지로 프로그램에 내장할 수 있는 경량 오픈소스 DBMS이다.

#### 4. IoT 데이터의 통계적 처리 기능

III장에서 설계한 내용을 바탕으로 Fig. 32와 같은 사용자 인터페이스를 구현하였다. 이 인터페이스를 이용하면, 원하는 데이터를 사용자 친화적인 규칙을 이용하여 조회할 수 있을 뿐만 아니라, 선택한 데이터들의 시간 속성이 서로 상이하여 상관관계 비교가 어려울 경우에 원시 데이터 대신 상황에 맞는 기초통계 데이터를 가져오도록 함으로써, 추가적인 데이터 변환 과정 없이도 간단하게 탐색적 데이터 분석을 수행할 수 있게 구현하였다.

표출되는 데이터들은 라인 그래프 형식으로 시각화하여, 시계열 분석이 가능하도록 구현하였고, 원하는 데이터 소스를 선택한 후 비교 대상 데이터 소스를 선택 또는 해제하면서 분석 작업을 진행할 수 있게 하였다. 또한, 기간 범위를 드래그 앤드 드롭 방식으로 유연하게 변경 가능하도록 구현하였다.

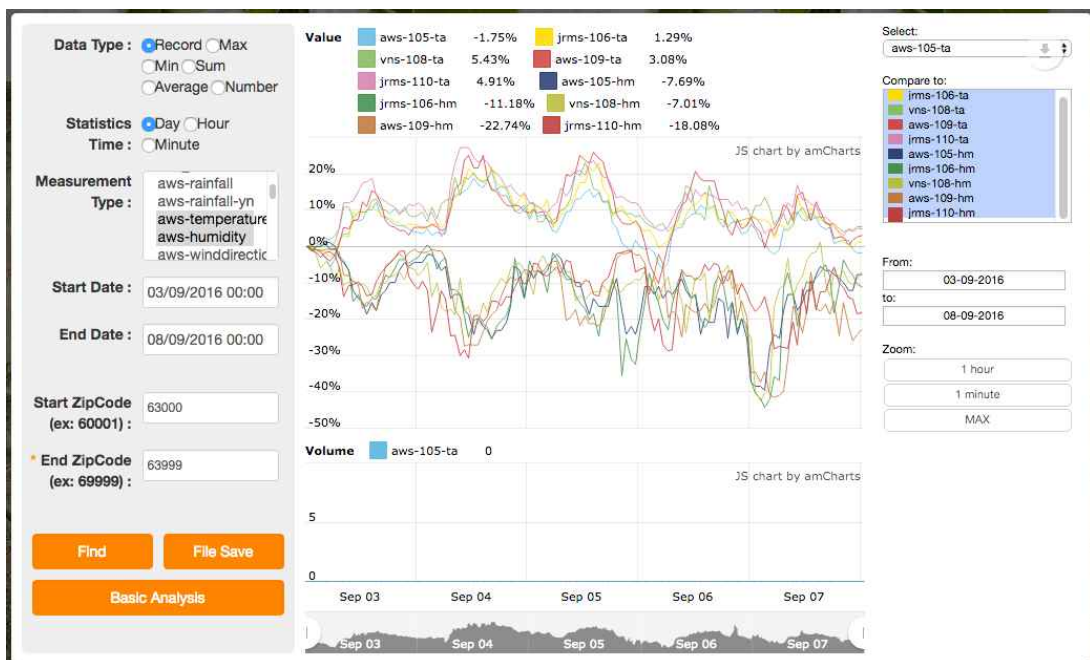


Fig. 32. UI for data analysis

이렇게 조회하고, 선택한 데이터들은 추가적인 분석 작업에 활용할 수 있도록 데이

터 세트로 출력할 수 있게 하였는데, 통계 분석 작업에 쉽게 사용할 수 있도록 Fig. 33과 같이 행렬 형식으로의 출력 기능을 구현하였다.

```
File Name : StatisticsFile_1479432828899.csv

File Size : 59177.0

File Content : date, 138, 144, 150, 156, 162, 139, 145, 151, 157, 163
2016-09-03 00:00:00.0, 23.0, 23.4, 22.4, 22.6, 23.0, 91.0, 93.5, 98.5, 96.5, 97.5
2016-09-03 00:10:00.0, 23.0, 23.4, 22.3, 22.6, 23.0, 91.0, 93.3, 98.0, 96.5, 97.4
2016-09-03 00:20:00.0, 22.9, 23.4, 22.3, 22.6, 22.8, 91.0, 93.0, 97.6, 96.4, 97.2
2016-09-03 00:30:00.0, 22.9, 23.3, 22.2, 22.6, 22.7, 91.0, 93.0, 97.9, 96.4, 97.4
2016-09-03 00:40:00.0, 22.9, 23.4, 22.2, 22.7, 22.5, 90.0, 92.8, 98.5, 96.4, 97.6
2016-09-03 00:50:00.0, 22.9, 23.3, 22.1, 22.7, 22.4, 91.0, 93.0, 98.5, 96.3, 97.9
2016-09-03 01:00:00.0, 23.0, 23.3, 22.0, 22.7, 22.5, 90.0, 93.2, 99.0, 96.2, 97.7
2016-09-03 01:10:00.0, 23.0, 23.3, 22.0, 22.7, 22.5, 90.0, 93.2, 99.0, 96.1, 97.4
2016-09-03 01:20:00.0, 22.9, 23.3, 22.0, 22.7, 22.6, 90.0, 93.2, 98.3, 95.8, 97.1
2016-09-03 01:30:00.0, 22.9, 23.2, 22.0, 22.6, 22.4, 90.0, 93.0, 98.6, 95.7, 96.9
2016-09-03 01:40:00.0, 22.9, 23.2, 21.9, 22.6, 22.5, 90.0, 93.0, 98.6, 95.7, 96.9
2016-09-03 01:50:00.0, 22.9, 23.2, 21.8, 22.5, 22.5, 90.0, 93.0, 98.7, 95.6, 96.8
2016-09-03 02:00:00.0, 22.9, 23.2, 21.7, 22.5, 22.8, 90.0, 93.1, 98.7, 95.7, 96.7
2016-09-03 02:10:00.0, 22.9, 23.2, 21.7, 22.4, 22.5, 90.0, 93.1, 99.0, 95.8, 96.2

Created : 2016-11-18 10:34:18

Updated : 2016-11-18 10:34:18
```

Fig. 33. Matrix style dataset output

Fig. 33은 플랫폼 화면상에서 출력결과를 조회한 것이고, 이를 파일로 다운로드 받아 열어보면 실제로는 CSV(Comma-Separated Values) 파일 형식으로 저장되어 있음을 알 수 있으며, 그 내용은 Fig. 34와 같다.

	A	B	C	D	E	F	G	H	I	J	K
1	date	138	144	150	156	162	139	145	151	157	163
2	2016.9.3 0:00	23	23.4	22.4	22.6	23	91	93.5	98.5	96.5	97.5
3	2016.9.3 0:10	23	23.4	22.3	22.6	23	91	93.3	98	96.5	97.4
4	2016.9.3 0:20	22.9	23.4	22.3	22.6	22.8	91	93	97.6	96.4	97.2
5	2016.9.3 0:30	22.9	23.3	22.2	22.6	22.7	91	93	97.9	96.4	97.4
6	2016.9.3 0:40	22.9	23.4	22.2	22.7	22.5	90	92.8	98.5	96.4	97.6
7	2016.9.3 0:50	22.9	23.3	22.1	22.7	22.4	91	93	98.5	96.3	97.9
8	2016.9.3 1:00	23	23.3	22	22.7	22.5	90	93.2	99	96.2	97.7
9	2016.9.3 1:10	23	23.3	22	22.7	22.5	90	93.2	99	96.1	97.4
10	2016.9.3 1:20	22.9	23.3	22	22.7	22.6	90	93.2	98.3	95.8	97.1
11	2016.9.3 1:30	22.9	23.2	22	22.6	22.4	90	93	98.6	95.7	96.9
12	2016.9.3 1:40	22.9	23.2	21.9	22.6	22.5	90	93	98.6	95.7	96.9
13	2016.9.3 1:50	22.9	23.2	21.8	22.5	22.5	90	93	98.7	95.6	96.8
14	2016.9.3 2:00	22.9	23.2	21.7	22.5	22.8	90	93.1	98.7	95.7	96.7
15	2016.9.3 2:10	22.9	23.2	21.7	22.4	22.5	90	93.1	99	95.8	96.2
16	2016.9.3 2:20	22.8	23.2	21.6	22.4	22.5	90	93.1	98.6	95.9	96.8
17	2016.9.3 2:30	22.8	23.2	21.7	22.5	22.7	91	92.8	97.7	95.8	96.8
18	2016.9.3 2:40	22.8	23.2	22.1	22.4	22.7	90	92.9	97.2	95.7	96
19	2016.9.3 2:50	22.7	23.1	22.1	22.5	22.6	91	92.9	96.1	95.5	96.3
20	2016.9.3 3:00	22.7	23.1	22.5	22.4	22.7	91	92.9	95.3	95.4	96.3

Fig. 34. CSV file of dataset

이렇게 데이터 분석을 위해 출력한 데이터 세트를 이용하여 기초적인 분석 기법들을 일괄적으로 자동 수행하고 그 결과를 보고서로 출력해주는 기능을 구현하였다. 이 때, 통계 전용 프로그래밍 언어인 R[43]을 이용하였고, 플랫폼 상에서 R 언어와 연동되어 프로그램이 수행되도록 구현하였다. Fig. 35는 기초 분석 자동 수행을 위한 R 프로그래밍 스크립트의 일부분이다.

```

#기초분석
library(gridExtra);
library(grid);
library(gtable);
library(ggplot2);
library(reshape);
library(dplyr);
library(corrplot);
library(xtable);

dir_name<-setwd("./");
data_file<-"aquaData.csv";
result_file<-"analysis.pdf";

data_file<-paste(dir_name,data_file,sep="/");

DataFile <- read.csv(data_file);

... 중략 ...

par(mfrow=c(2,2),mar=c(4, 3, 3, 1),oma=c(0.5, 0.5, 2, 0.5));
plot(lm.data);

dev.off();

```

Fig. 35. R program code for basic analysis

## V. 실험 및 평가

본 논문에서 사물인터넷 서비스 플랫폼 확장을 통해 이루고자 하는 목표는 데이터 활용성 강화이다. 이를 위해 데이터 품질기준을 참조하여 유용성, 접근성, 적시성을 각각 만족시킬 수 있는 방법들을 제안하고 구현하였다. 이 장에서는 이렇게 구현된 각 기능들이 정확하게 동작하는지 여부를 실험하였다.

시스템에 대한 실험은 세 가지 주제로 구분하여 수행하였다. 먼저, 유용성 확보를 위해 제안한 IoT 데이터 연합을 구성하는 기능 중 핵심 구성요소인 DFA가 설정 변경에 적절하게 반응하는 지를 실험하였다.

이어서 접근성 확보를 위해 개발한 육하원칙 기반 사물인터넷 데이터 검색 기능을 이용하여 사용자 친화적인 방식으로 통합 질의를 수행하고 정확한 결과를 반환하는 지 실험하였다.

그리고 마지막으로 적시성 확보를 위해 개발한 사물인터넷 데이터 통계적 처리 기능을 이용하여 자동 기초 분석을 수행하고 적시에 결과를 반환하는 지 실험하였다.

### 1. DFA의 데이터 변환

먼저, DFA가 설계 시 의도된 대로 동작하는 지 실험하였다. 실험은 DFA가 자동으로 DFMS로부터 설정 정보를 읽어 들이고, 연합 대상 이종 시스템들의 다양한 데이터 저장환경에 동적으로 적응하며, 정확하게 데이터를 조회하여 표준 형식으로 변환하는 동작을 수행하는지 확인하는 과정으로 구성하였다. 실험 개요는 Table 11과 같다.

이 실험의 핵심은 DFA가 소스코드나 설정파일 수정 없이 DFMS에서 제공하는 다양한 설정정보들을 정확하게 반영할 수 있는 지 확인하는 것이므로, 하나의



시스템에 DFA를 설치해두고 DFA가 참조하는 DFMS에 저장되어 있는 연합 대상 이종 시스템의 설정정보를 순차적으로 변경하였다. 이 때 변경되는 정보는 각기 다른 종류의 DBMS와 스키마로 조합된 3종의 설정 세트이다.

Fig. 36에서 보듯이 DFA가 변경되는 설정정보를 적시에 읽어 들여 사전 정의된 표준 형식에 맞게 데이터를 변환한 후 내장 DB에 정상적으로 저장하는 것을 확인하였다. 이는 데이터 연합 구성이 완료되었음을 의미하며, 이렇게 저장된 데이터를 통해 사용자의 통합 질의 명령에 신속하게 데이터를 반환할 수 있는 상태가 되었음을 확인할 수 있다.

Table 11. Data translation testing overview

job	description
prepare three experimental legacy databases	schema :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 changing the sensor information	experimental bus information system data transformed result experimental parking information system data transformed result experimental water information system data transformed result

이 실험을 통해 확인할 수 있는 점은 각기 다른 운영체제, DBMS, DB 스키마로 조합된 환경에 동적으로 적용할 수 있는 범용 에이전트 프로그램을 이용하여 이종 시스템 데이터를 효과적으로 통합 활용할 수 있게 된다는 것이다.

이 방식을 이용하면, 기존 기술을 이용하여 데이터 연합을 구성하고자 할 때 진행되는 과정과 비교할 때, 통합 데이터 모델을 개발하는 작업과 시스템별 연동 프로그램을 개발(커스터마이징 포함)하는 과정을 생략할 수 있게 되므로 결과적으로 데이터 활용성의 부특성 중 유용성의 하위 특성인 데이터의 충분성을 기존



기술에 비해 신속하게 향상시킬 수 있게 된다.

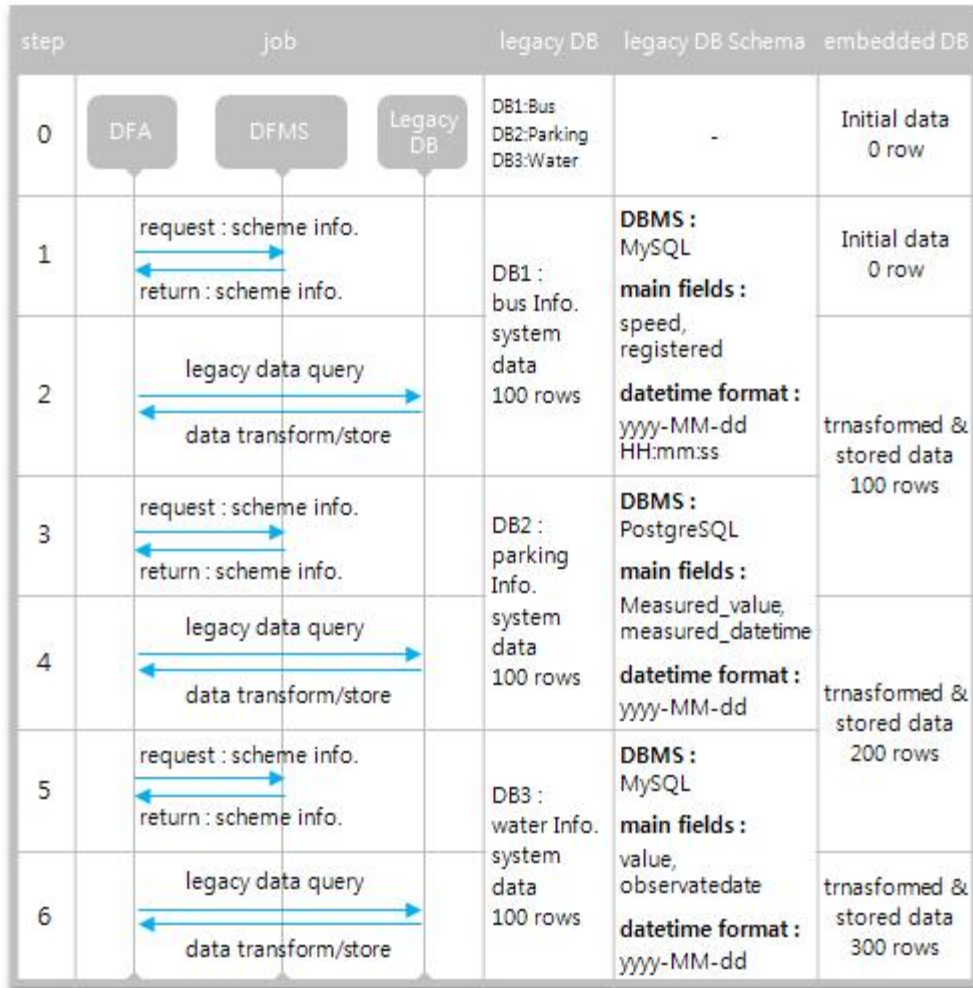


Fig. 36. Flow of data translation test

## 2. 통합 질의

실험을 통해 DFA가 정확하게 동작하고, 데이터 연합 구성이 완료되었음을 확인하였다. 이에 사용자 관점에서 어떤 데이터가 어느 시스템에 존재하는지, 데이터베이스 구조가 어떠한지 알지 못하더라도, 사용자가 이해하기 쉬운 어휘와 기준을 이용하여 통합 질의를 수행하는 과정에 대한 실험을 추가로 진행하였다. 이는 데이터 활용성의 부특성 중 접근성을 향상시키기 위해 제안된 기능이다.

Table 12. Federated systems for union query

Legacy Systems	Description
System #1 (sensorId : 1)	<ul style="list-style-type: none"> <li>OS : Windows Server 2008</li> <li>DBMS : MySQL 5</li> <li>Data measurementType : Waterlevel measurementDatetime : 20150630~20151231 measurementZipcode : 50000</li> </ul>
System #2 (sensorId : 2)	<ul style="list-style-type: none"> <li>OS : Cent OS 7</li> <li>DBMS : MySQL 5</li> <li>Data measurementType : Rainfall measurementDatetime : 20150930~20160331 measurementZipcode : 50001</li> </ul>
System #3 (sensorId : 3)	<ul style="list-style-type: none"> <li>OS : Mac OS X 10.11</li> <li>DBMS : PostgreSQL 9.4</li> <li>Data measurementType : Rainfall measurementDatetime : 20160101~20160630 measurementZipcode : 50002</li> </ul>

Table 12와 같이 실시간 통합 질의 대상이 되는 레거시 시스템들을 실험용으로 구성하고 DFMS를 이용하여 조건을 변경해가면서 데이터 조회를 시도하였다.

용이한 결과 확인을 위해 레거시 시스템 1대당 보유 센서 1개씩의 센서 데이터가 DFMS에 연결되도록 구성하였고, 센서 식별자는 레거시 시스템 번호와 같게 부여하였다.

Fig. 37은 조건별 데이터 조회 기능을 제공하는 통합 질의용 사용자 인터페이스 구현 결과이며, 이를 이용하여 실험을 진행했다.

Fig. 37. Union query settings

통합 질의 실험에 사용한 조건들과 실제 수행결과를 Table 13에 정리하였다. 여기에서 제시된 조건과 결과들은 다음과 같은 의미를 갖는다. Test 1은 아무 제약 없이 모든 레거시 시스템으로부터 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 상관없이 DFMS에서 새롭게 부여한 식별자이다. 따라서 조건 변경에 따라 어느 레거시 시스템에서 데이터를 가져오는 지 확

인하기 위해서 각 결과에 나타난 'sensorId'를 확인하였다.

Test 2는 측정유형을 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 측정일시 및 측정위치를 제약으로 사용하였고 조건에 부합하는 데이터가 해당 시스템으로부터 반환되었음을 확인하였다.

이 실험에서 사용한 쿼리문들을 기존 시스템에서 동일하게 수행하고자 한다면, 분산된 데이터를 각각 찾아낼 수 있도록 시스템별로 다른 구조를 학습하는 과정이 필요하고, 시스템의 수만큼 쿼리문을 작성해야 하며, 실험에서 검색 기준으로 활용한 What, When, Where에 대한 검사를 하나의 쿼리문으로 작성할 수 없기 때문에 두 번 또는 세 번에 나눠 수행해야만 할 것이다.

이러한 점에 비춰볼 때, 본 논문에서 제안한 방법을 사용할 경우 작업의 수를 단축시키는 효과와, 질의 수행 루틴의 복잡성을 해소하는 효과가 발생하는 것을 확인할 수 있다.

Table 13. Results of union queries

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

### 3. 자동 기초 분석

IV장에서 구현한 통계활용 지원기능을 이용하여 자동기초분석을 수행하고, 그 결과를 확인하였다. 이 실험의 목적은 데이터 활용성의 부특성 중 하나인 즉시성 향상 여부에 대한 확인이다. 본 논문에서는 원시데이터에 대한 기초통계 데이터의 실시간 생성 및 활용을 제안하였는데, 이러한 과정을 통해 별도의 전처리 과정을 추가적으로 수행하지 않고도 다음과 같이 다양한 관점에서의 기초분석 결과를 원하는 때에 즉시 획득할 수 있게 된다.

Fig. 38-43은 행렬 형태로 출력한 데이터 세트를 활용하여 자동기초분석을 수행한 결과로서, 각각 기초통계량, 박스플롯, 시계열플롯, 상관플롯, 스캐터플롯, 회귀분석에 대한 결과표 및 그래프이다.

Analysis Data Infomation					
2	nitrate	salinity	food	temp	ph
437	Min. :-1.52859	Min. :-0.3748	Min. :-2.26765	Min. :-2.4126	Min. :-1.96246
7286	1st Qu.:-0.64749	1st Qu.:-0.3748	1st Qu.:-0.07972	1st Qu.:-1.0084	1st Qu.:-0.86441
1724	Median :-0.03749	Median :-0.3748	Median : 0.33052	Median : 0.7469	Median : 0.01402
1000	Mean : 0.00000	Mean : 0.0000	Mean : 0.00000	Mean : 0.0000	Mean : 0.00000
1248	3rd Qu.: 0.43695	3rd Qu.:-0.3748	3rd Qu.: 0.60401	3rd Qu.: 0.7469	3rd Qu.: 0.61794
414	Max. : 2.67359	Max. : 2.7335	Max. : 1.09629	Max. : 1.8001	Max. : 2.86892

Fig. 38. Output of basic statistics calculation

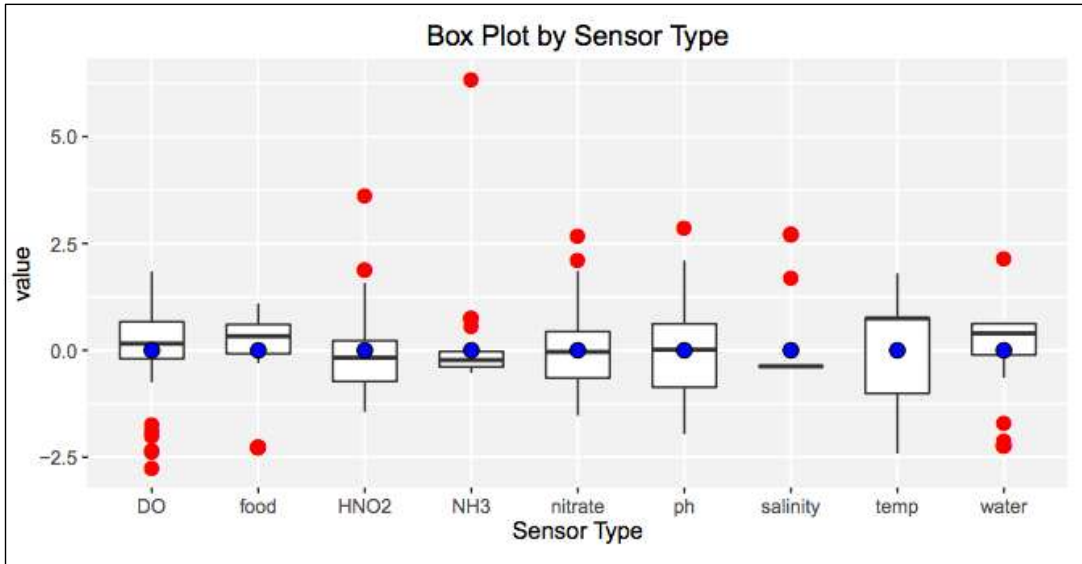


Fig. 39. Box plot chart

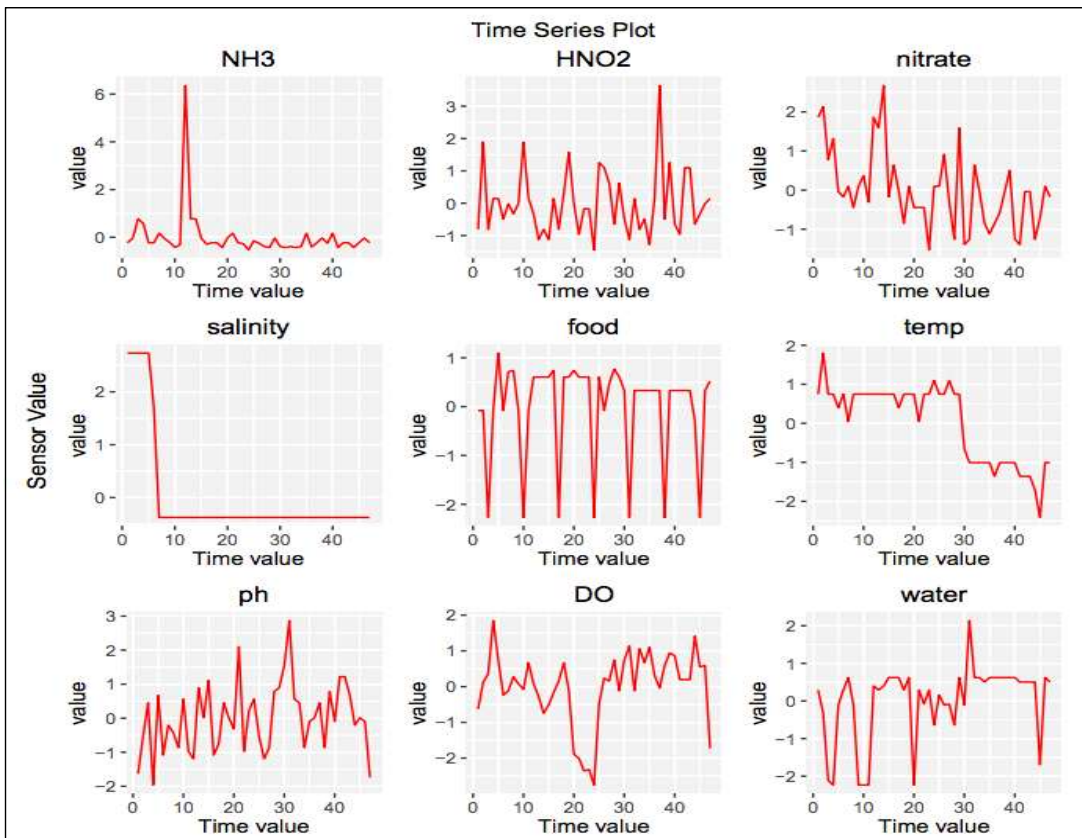


Fig. 40. Time series plot chart



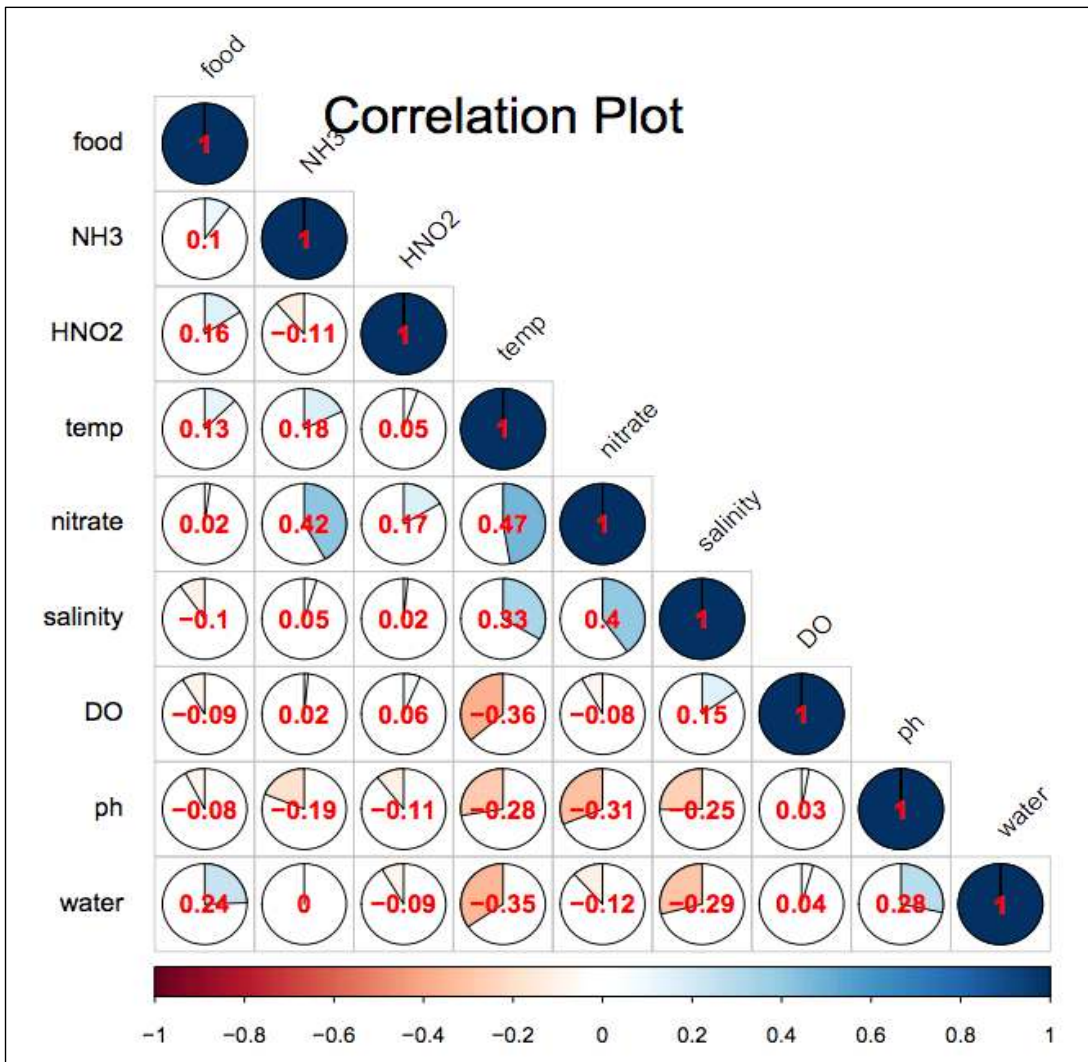


Fig. 41. Correlation plot chart



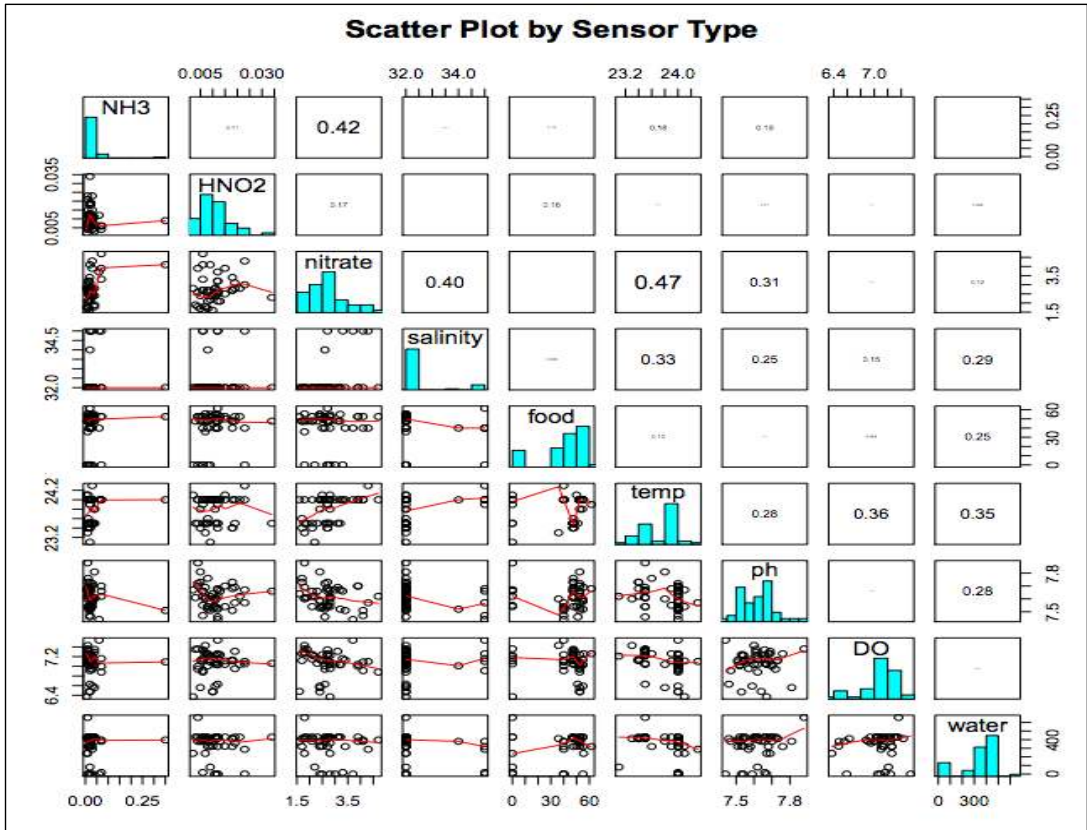


Fig. 42. Scatter plot chart

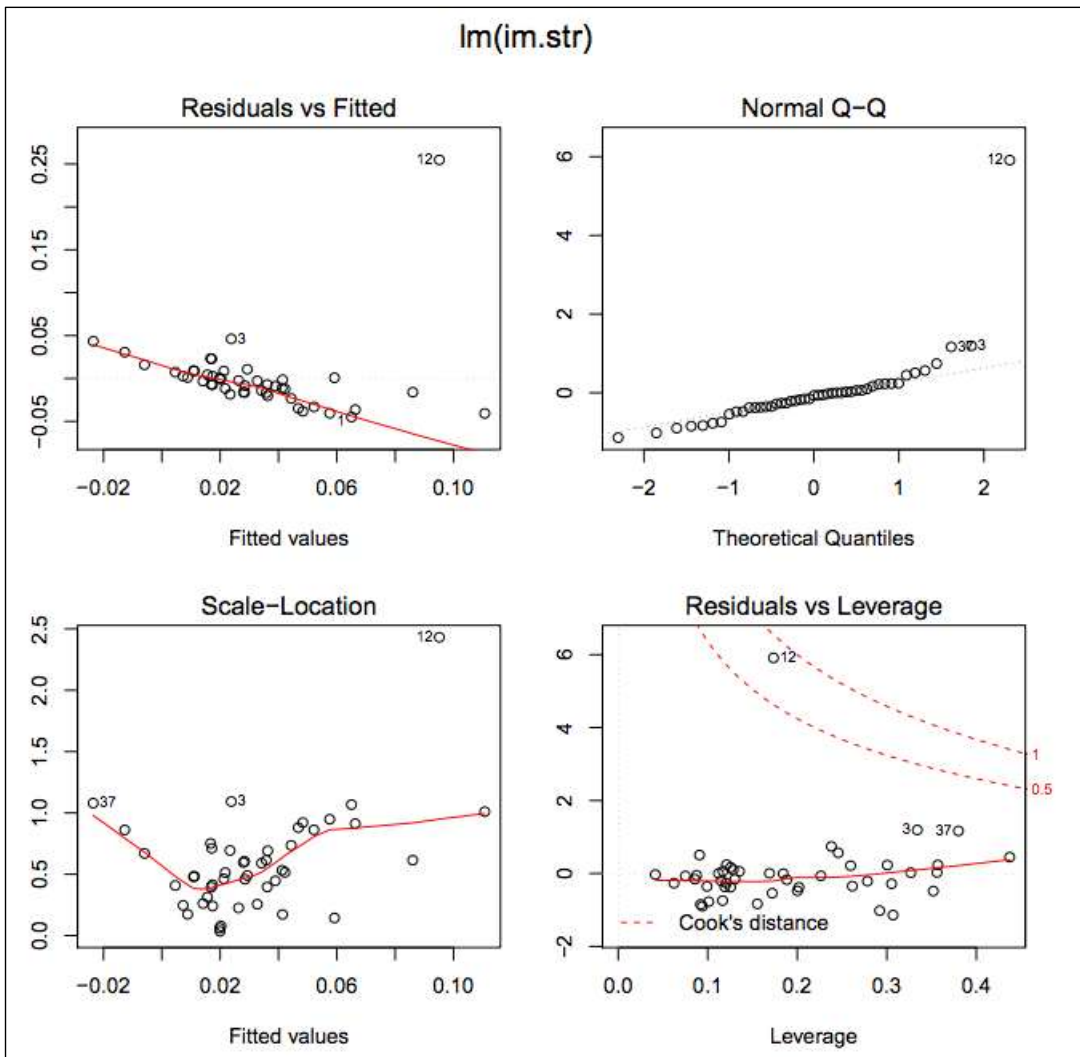


Fig. 43. Regression analysis

이 결과들을 얻기 위해 필요한 일련의 작업들은 모두 자동방식으로 수행되었는데, 이는 기존의 시스템과 비교할 때 편의성 향상 이상의 의미를 갖는다. 전처리 과정이 실시간으로 수행된 상태를 유지하기 때문에 사용자가 요구하는 데이터를 즉시 제공할 수 있다는 점이 그것이다.

#### 4. 실험 결과 평가

이러한 실험을 통해 확인할 수 있는 기존 연구와의 차별성은 다음과 같다.

첫째, oneM2M 표준[5]이 레거시 시스템 연동을 위해 기존 장치관리 표준기술들에 대한 인터페이스를 제공하는 것과 달리, 본 연구에서 제안하는 모델은 연동하고자 하는 레거시 시스템이 비표준 기술을 사용하더라도, 연합 데이터베이스 구성 기능을 이용하여 그 데이터를 용이하게 활용할 수 있음을 확인하였다. 뿐만 아니라, 제안하는 모델은 CUAHSI ODM[26]의 관측 데이터 모델링 아이디어를 바탕으로 사물인터넷 환경에 적합한 데이터 관리 형식을 사전에 정의하여 제공함으로써, JBoss Data Virtualization[31]을 이용할 때 필요한 과정인 ‘목표 데이터세트의 형식 정의 작업’을 생략할 수 있으며, 그럼에도 정상적으로 데이터 통합과 활용이 가능함을 확인하였다.

둘째, 대표적인 사물인터넷 서비스 플랫폼 기술인 oneM2M 표준에서 제공하는 데이터 검색 기준을 확대하여 적용함으로써, 서비스 플랫폼을 통해 수집된 데이터를 육하원칙과 같은 사용자 친화적 필터 규칙을 이용하여 검색하는 것이 가능함을 확인하였다.

셋째, 다수의 이종 시스템으로부터 수집된 다양한 형태의 데이터들에 대해 별도의 전처리 작업 없이도 즉시적, 반복적으로 탐험적 분석을 수행하고, 이어서 자동적인 방법으로 기초 통계 분석까지 수행할 수 있음을 확인하였다.

본 논문에서 이루고자 하였던 연구 목표는 기존 기술의 기능적 확장을 통해 데이터 활용성을 강화하는 것이었다. 여기서 데이터 활용성 강화란 데이터 품질 특성으로 정의되는 속성들에 대한 향상을 의미하지만, 궁극적으로는 가용한 데이터를 바탕으로 융합 서비스 개발과 같은 업무를 수행하기 위해 거쳐야 하는 작업의 수를 줄이는 것을 의미하기도 한다.

Table 14. Data integration and utilization procedures

Step	Detailed Procedure	Whether to Perform		Operation Difficulty	Workload
		Legacy	Expanded		
1. Legacy System Integration	1-1. Analyzing the data handling and structure of the systems to be integrated	O	O	Normal	It is repeated as many times as the number of systems to be integrated
	1-2. Building an integrated data model (global view)	O	Unnecessary	High	one time
	1-3. Developing data integration programs for the systems to be integrated	O		High	It is repeated as many times as the number of systems to be integrated
	1-4. Mapping the data structure of each system	O	O	Low	It is repeated as many times as the number of systems to be integrated
2. Data Query	2-1. Executing union query based on 5W1H	Not available	O	Low	one time
	2-2. Querying for data(What, Who, Why, How) -> A list of things is returned	O	Unnecessary	Low	It is repeated as many times as the number of systems to be integrated
	2-3. Extracting a query that examines the positional conditions of the things in the result of the step 2-2 -> A list of things is returned	O		Low	It is repeated as many as the number of things

	2-4. Extracting a query that retrieves a time-based value data of the things contained in the results of the step 2-3	O		Low	It is repeated as many times as the number of things multiplied by the number of measurement times
3. Data Analysis	3-1. Value unit unification	O	Unnecessary	Low	It is repeated as many times as the number of attributes of the data
	3-2. Time format unification	O		Low	
	3-3. Time period unification	O		High	
	3-4. Matrix type dataset generation	O	△ (Automatic)	Low	one time
	3-5. Basic statistics Analysis	O	△ (Automatic)	Normal	It is repeated as many as the number of analysis types

Table 14는 사물인터넷 서비스 플랫폼에서 다중 도메인에 걸쳐 융합 서비스를 개발하고자 할 때 필요한 작업들의 절차를 나타내고 있으며, 본 논문에서 확장 방안으로 제시하였던 세 가지 기능과 그 세부 작업들에 대해서 기존 방식과 확장된 방식의 수행여부 차이를 표현한다.

총 3단계 절차와 13개의 작업을 제시하고 있으며, 기존 방식은 12개의 작업을 수행해야 하고, 확장된 방식은 5개의 작업만 수행하면 되는 것으로 구분되었다. 작업의 수 기준으로는 약 60% 단축이 가능한 것이며, 확장된 방식에서 수행하는 5개의 작업들은 상대적으로 난이도가 낮고 작업량이 적은 항목이 대부분이라는 점을 감안하면 실질적으로는 기존 대비 90% 이상 작업 단축이 가능할 것으로

판단된다.

또한, 본 논문에서 제안한 기능들은 서론 및 관련연구를 통해 제시하였던 데이터 활용성의 부 특성 네 가지 중 세 가지의 수준을 향상시킬 수 있음을 확인하였다. 이러한 결과는 본 논문의 목표인 ‘데이터 활용성 강화’를 위한 기능들이 의도한대로 구현되었음을 의미하는 것으로 해석할 수 있다.

## VI. 결 론

본 연구에서는 수평적 서비스 구조를 갖는 사물인터넷 서비스 플랫폼인 oneM2M 표준을 이용하여 이종 도메인/시스템 간 자원 공유 및 서비스 융합이 실무적으로 가능하도록 육하원칙 기반의 자원 관리 모델을 제안하였다. 이를 위해, oneM2M 표준에서 제시하는 <Filter Criteria>를 기준으로 확장해야 할 요소를 도출하고, 육하원칙 질문요소를 사물인터넷 구성요소에 사상(Mapping)하는 작업을 수행하였다. 또한, 확장된 <Filter Criteria>의 수용이 가능하도록 표현력이 향상된 자원 관리 모델을 제안하였다. 마지막으로 이렇게 도출된 <Filter Criteria>와 자원 관리 모델 하에서 육하원칙 기반의 자원 검색이 수행되는 절차를 설명하였다.

그리고, 연합 데이터베이스 모델을 활용하여 다양한 운영체제, DBMS 종류, DB 스키마 구조를 갖는 이종 시스템들을 논리적으로 가상 통합하여 통합 질의를 수행할 수 있는 모델을 제시하였고, 수집되는 데이터를 통계분석에 용이하게 활용할 수 있는 방안을 제시하였다.

또한, 이러한 개념들에 대해 설계 및 구현을 완료하였고, 제안한 모델의 기능을 검증하기 위해서 실험을 수행하였다.

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

그리고, 실험용 이종 시스템들과 DFMS를 이용하여 조건을 변경해가며 통합 질의 기능을 실험하였다. 이 실험을 통해 물리적으로 분리되어 있는 이종 시스템의 이종 데이터들을 가상으로 통합하여 측정일시, 측정위치, 측정유형 요소 지정만으로 데이터를 통합 검색할 수 있음을 확인하였고, 수집된 데이터의 탐험적 분석 및 자동 기초 분석이 가능함을 확인하였다.

이러한 모델을 활용하면 보다 용이하게 사물인터넷 데이터를 활용할 수 있으며, 이를 통한 데이터 융합 범위 확대 및 응용 서비스 고도화를 기대할 수 있다.

본 연구의 결과는 향후 oneM2M 표준과 같은 개방형 사물인터넷 서비스 플랫폼의 실무적 구현 시에 자원 공유 및 서비스 융합 요구에 대한 확장 방안으로써 활용될 수 있을 것으로 기대된다.

다만, 본 연구에서는 기존 oneM2M 표준의 ‘아키텍처 규격’을 기준으로 확장 모델을 제시했으나, 현재 oneM2M의 시맨틱 기술 지원방안에 대한 표준 개발이 진행되고 있다는 점을 고려할 때, 본 연구 주제가 시맨틱 기술의 관점에서 다시 한 번 검토 및 연구되어야 할 필요가 있다고 하겠다. 또한, 육하원칙 기반으로 사용자 친화적인 데이터 검색 기능을 제시하였으나, 사용자 관점에서 어떤 데이터가 플랫폼 내에 분포하고 있는지를 검색 전에 파악할 수 있도록 시간, 유형, 공간 기준으로 데이터 분포 현황을 개략적으로 제시할 수 있는 방안에 대한 연구가 이루어진다면 본 논문에서 제시한 IoT 데이터 검색 기능의 효용성이 증대될 수 있을 것이다.

그리고, 이종의 레거시 시스템 연동기능의 경우 현재는 RDBMS에 저장된 데이터를 통합하는 기능만을 고려하였으나, 향후, DFA의 기능 확장을 통해 NoSQL, File, Hadoop Storage 형식의 데이터를 연동하는 기능을 제공하는 방법도 연구할 필요가 있다.

데이터 분석 기능의 경우, 현재는 데이터 간의 상관관계 분석에 초점을 맞춰 구현하였으나, 향후, 보다 다양한 데이터 마이닝 기법 활용이 가능한 구조로의 개선에 대한 추가 연구가 필요하다.



## 참 고 문 헌

- [1] Internet of Things cases - 15 kinds of smart IoT Products & Services,  
<http://smartincome.tistory.com/250>
- [2] IoT (Internet of Things) products and services, Trend 2015,  
<http://www.slideshare.net/firehj/2015-56280477>
- [3] Internet of Things products 73 kinds of review materials,  
<http://www.slideshare.net/kimbongzo/internet-of-things-iot-solution-73-73>
- [4] S.G. Hong, H. Lee, J.C. Choi, 2015, M.N. Bae, K.B. Lee, Internet of Things Software Platforms Technology Trends, Electronics and Telecommunications Trends, Vol. 30, No. 5, pp.49-58
- [5] oneM2M, <http://www.onem2m.org/>
- [6] AllJoyn, <https://allseenalliance.org/framework>
- [7] IoTivity, <https://www.iotivity.org/>
- [8] Interoperability Standardization of oneM2M and industry-standard,  
<http://www.industrysolutions.co.kr/>
- [9] Data Quality Certification, <https://www.dqc.or.kr/guideline/1-1-1.html>
- [10] D, Heimbigner, D. McLeod, 1985, A Federated Architecture for information management, ACM Transactions on Office Information Systems, Vol. 3, No. 3, pp.253-278
- [11] A. P. Sheth, J. A. Larson, 1990, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol. 22, No. 3, pp.183-236
- [12] Y.K. Kim, G.W. Lee, R.I. Heinsen, E.N. Huh, 2016, A Study on Data Storage Federation Model for Increasing Data Availability, Paper presented at Korea Computer Congress 2016, pp.1270-1272
- [13] Federated database system,

- [https://en.wikipedia.org/wiki/Federated\\_database\\_system](https://en.wikipedia.org/wiki/Federated_database_system)*
- [14] G.E. Park, C.J. Kim, 2015, Quality Characteristics of Public Open Data, Journal of Digital Convergence, Vol. 13. No. 10, pp.135-146
- [15] Functional Architecture(TS-0001-V1.13.1), oneM2M, February 2016
- [16] Five Ws, *[https://en.wikipedia.org/wiki/Five\\_Ws](https://en.wikipedia.org/wiki/Five_Ws)*
- [17] S.C. Choi, N.M. Seong, J.S. Yoon, J.H. Kim, 2015, Internet of Things platform open source trends, The Journal of The Korean Institute of Communication Sciences, Vol. 32, No. 5, pp.16-22
- [18] CoAP(Constrained Application Protocol), *<http://coap.technology/>*
- [19] MQTT(MQ Telemetry Transport), *<http://mqtt.org/>*
- [20] J.H. Kim, I.G. Oh, J.H. Han, G.B. Kim, G.T. Chang, J.P. Youn, 2016, Developing differentiated functionalities of IoT platform through implementing extended oneM2M standards, Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp.80-82
- [21] J. G. Song, S. S. Kim, 2006, Context-based Data Modeling for Ubiquitous Computing Information Management, Vol. 6, No. 3, pp.55-62
- [22] J. C. Augusto, J. O'Donoghue, 2009, CONTEXT-AWARE AGENTS The 6Ws Architecture, Proceedings of the International Conference on Agents and Artificial Intelligence, Porto, Portugal, pp.591-594.
- [23] BDI(Belief - desire - intention software model),  
*[https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention\\_software\\_model](https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model)*
- [24] J. S. Son, 2016, IoT-CBDAC: Context-based Dynamic Access Control Model using Intuitive 5W1H for the Internet of Things, Dissertation for the Degree of Doctor, Department of Computer and Radio Communications Engineering, Graduate School Korea University
- [25] Consortium of Universities for the Advancement of Hydrologic Science, Inc.,  
*<https://www.cuahsi.org/>*
- [26] D.G. Tarboton, J.S. Horsburgh, D.R. Maidment, 2008, CUAHSI Community

- Observations Data Model(ODM) Version 1.1 Design Specifications, CUAHSI
- [27] OMA DM,  
*<http://openmobilealliance.org/about-oma/work-program/device-management/>*
- [28] BBF TR-069, *<https://www.broadband-forum.org/cwmp.php>*
- [29] OMA Lightweight M2M,  
*<http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0>*
- [30] 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, 2014, oneM2M service platform standard commentary, Telecommunications Technology Association, pp.121-132
- [31] Red Hat JBoss Data Virtualization,  
*<https://www.redhat.com/ko/technologies/jboss-middleware/data-virtualization>*
- [32] NoSQL(Not Only SQL), *<https://en.wikipedia.org/wiki/NoSQL>*
- [33] SaaS(Software as a service), *[https://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Software_as_a_service)*
- [34] Hadoop, *<http://hadoop.apache.org/>*
- [35] U.C. Park, 2007, Design of Database Integration System and Query System based on Global View Generation Tool, Journal of Korean Society for Internet Information, Vol. 8, No. 3. pp.65-74
- [36] unity JDBC, *<http://www.unityjdbc.com/>*
- [37] S.H. Min, H.Y. Kwak, S.J. Lee, 2016, 5W1H based resource management model for the expansion of oneM2M standard Data Filter rules, Journal of The Korea Society of Computer and Information, Vol. 21, No. 9. pp.85-90
- [38] B.K. Ahn, A Data Standardization Method for DB Integration, 2010, Master's Thesis, Graduate School of Information Sciences, Soongsil University
- [39] D.H. Suh, Chang-Keun Ryu, 2011, 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
- [40] E.H. Lee, 2012, 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

- [41] J.U. Hong, 2014, A study On Data Fusion Using Statistical Matching, Master's Thesis, Graduate School, Sungkyunkwan University
- [42] S.H. Min, H.Y. Kwak, S.J. Lee, 2016, Real-time Virtual Integration of heterogeneous system and Union Query System, Journal of The Korea Society of Computer and Information, Vol. 21, No. 10. pp.107-114
- [43] The R Project for Statistical Computing, <https://www.r-project.org/>
- [44] Java Persistence API, [https://en.wikipedia.org/wiki/Java\\_Persistence\\_API](https://en.wikipedia.org/wiki/Java_Persistence_API)
- [45] Hibernate ORM, <http://hibernate.org/orm/>
- [46] Apache Derby, <https://db.apache.org/derby/>