

마스터 데이터를 활용한 마이크로 서비스 아키텍처에서의 데이터 베이스 리버스 엔지니어링

신광철¹ · 이춘열^{2*}

Database Reverse Engineering Using Master Data in Microservice Architecture

Kwang-chul Shin¹ · Choon Y. Lee^{2*}

¹Graduate Student, The Graduate of Business IT, Kookmin University, Seoul, 02707, Korea

^{2*}Professor, The Graduate of Business IT, Kookmin University, Seoul, 02707, Korea

요 약

마이크로 서비스 아키텍처는 매우 밀접한 비즈니스 기능을 수행하기 위한 목적으로 시스템을 소형 경량 서비스로 분할하는 것에 중점을 두고 있기 때문에 민첩성, 개발 생산성, 신뢰성, 배포 용이성에만 집중하는 경향이 있다. 마이크로 서비스 아키텍처에서 바라보는 데이터베이스는 단지 데이터를 저장하고 추출하는 파일 혹은 스토리지이며, 소프트웨어 개발 편리성과 확장성을 위해서 데이터 품질은 희생될 수 있음을 강조한다. 데이터베이스 구조와 데이터 의미를 이해하기 위해 사용되는 데이터베이스 리버스 엔지니어링은 비즈니스 의사 결정을 위한 데이터 활용을 위해 필요하다. 하지만, 데이터 품질을 경시하는 마이크로 서비스 아키텍처에서는 리버스 데이터베이스 엔지니어링 적용에 어려움이 있다. 본 연구는 해결책으로써 개념 데이터 모델 복원 시 마스터 데이터를 활용한 데이터 리버스 엔지니어링 방법을 제안한다. 적용 사례로 마이크로 서비스 아키텍처로 구현된 반쯤 서비스 데이터베이스에 제안 방법을 구현하여 적용 가능성을 검증하였다.

ABSTRACT

Microservice architecture focuses on dividing it into small and lightweight services to build for the purpose of performing very close business functions. So it tends to concentrate only on agility, productivity, reliability, and ease of deployment of software development. Microservice architecture considers database as just a file or storage for storing and extracting data, emphasizing that data quality can be sacrificed for convenience and scalability of software development. Database reverse engineering for understanding database structure and data semantics is needed for data utilization for business decision making. However, it is difficult that reverse database engineering is applied in microservice architecture that neglects data quality. This study proposes database reverse engineering method that utilizes master data to restore the conceptual data model as a solution. The proposed method is applied to the return service database implemented by microservice architecture and verified its applicability.

키워드 : 데이터베이스 리버스 엔지니어링, 데이터 모델링, 마스터 데이터, 마이크로 서비스 아키텍처

Keywords : Database Reverse Engineering, Master Data, Microservice Architecture, Data Modeling

Received 14 March 2019, Revised 26 March 2019, Accepted 26 March 2019

* Corresponding Author Choon Y. Lee(E-mail:cylee@kookmin.ac.kr, Tel:+82-2-910-4565)

Professor, The Graduate of Business IT, Kookmin University, Seoul, 02707, Korea

Open Access <http://doi.org/10.6109/jkiice.2019.23.5.523>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

요즘 서비스 기반 소프트웨어 산업에서 아키텍처 스타일을 선택할 때 마이크로 서비스 아키텍처가 대부분을 차지하고 있다. [1]에서 마이크로 서비스는 시스템을 매우 밀접한 비즈니스 기능을 수행하기 위한 목적으로 구축하는 소형 경량 서비스로 분할하는 것에 중점을 두고, 전통적인 서비스 지향 아키텍처 스타일의 진화된 형태로 정의하고 있다. 마이크로 서비스 아키텍처를 적용함으로써 민첩성, 개발 생산성, 탄력성, 확장성, 신뢰성, 배포 용이성 향상되지만, 적절한 서비스 식별, 데이터 공유, 보안, 성능은 문제가 될 수 있다. 마이크로 서비스 아키텍처에서 바라보는 데이터베이스는 단지 데이터를 저장하고 추출하는 파일 혹은 스토리지이며, 소프트웨어 개발 편리성과 확장성을 위해서 데이터 품질은 희생될 수 있다.[2]

비즈니스 의사결정을 위한 데이터 활용을 위해서는 데이터베이스 구조와 데이터 의미를 파악하기 위해 데이터베이스 리버스 엔지니어링(Database Reverse Engineering)이 필요하다. 그러나 마이크로 서비스 아키텍처에서는 특별히 데이터 모델링 단계가 없으며 애플리케이션에서 사용하는 질의(Query) 중심으로 데이터 구조를 설계한다[3-9]. 이 때문에 데이터베이스 리버스 엔지니어링 시에 데이터 구조와 의미 파악에 어려움이 있다. 본 연구는 이에 대한 해결책으로 데이터베이스 리버스 엔지니어링에서 개념 데이터 모델을 복원할 때 마스터 데이터를 활용하여 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장은 이론적 배경을 통해 마스터 데이터의 의미와 기존 데이터베이스 리버스 엔지니어링 연구의 한계점을 알아보고, 3장은 마스터 데이터를 활용한 데이터베이스 리버스 엔지니어링 방법에 대해 설명한다. 4장은 제안 방법 현실 적용 가능성을 적용 사례와 분석을 통해서 알아보고 5장에서는 결론과 향후 과제를 서술한다.

II. 이론적 배경

2.1. 마스터 데이터

기업 활동에서 가장 중심이 되는 대상물들은 대표적으로 Party (고객, 사원, 시민, 조직 등), Thing(제품, 부

품, 원재료, 기계, 건물, 창고 등)이 있으며, 이런 종류의 데이터는 개념 설계부터 매우 중요하게 다루어져야 한다. 특히 최근에 다양한 데이터베이스들과 정보시스템들이 기업이나 조직에서 활용되면서부터 핵심 데이터에 대한 중요성이 더욱 커지고 있는데, 이런 종류의 데이터를 마스터 데이터라고 한다 [10]. 마스터 데이터는 트랜잭션 데이터가 참조해야만 하므로 트랜잭션 데이터 품질을 점검할 수 있는 기준이 될 수 있다.

2.2. 데이터베이스 리버스 엔지니어링

데이터베이스 리버스 엔지니어링은 산출물이 존재하지 않거나 산출물을 갱신할 필요성이 있는 데이터베이스의 스키마, 데이터, 화면 등을 입력으로 하여 개념 데이터 모델 복원을 목적으로 한다.

데이터베이스 리버스 엔지니어링 단계는 [11]와 [12]에 상세히 언급되어 있다. [11]에서는 2단계로 데이터 구조 추출(Data Structure Extraction)과 데이터 구조 개념화(Data Structure Conceptualization)로 구분했고, [12]에서는 3단계로 구현 복구(Implementation Recovery), 설계 복구(Design Recovery) 그리고 분석 복구(Analysis Recovery)로 구분한다. 본 논문에서는 물리 데이터 모델 복구(Physical Data Model Recovery), 논리 데이터 모델 복구(Logical Data Model Recovery) 그리고 개념 데이터 모델 복구(Conceptual Data Model Recovery)의 단계로 구분하며 본 논문에서 마스터 데이터를 적용할 단계는 개념 모델 복구이다. 표 1은 데이터베이스 리버스 엔지니어링 단계를 비교 정리했다.

Table. 1 Database Reverse Engineering Steps

Hainaut[11]	Blaha[12]	This Paper
Data Structure Extraction	Implementation Recovery	Physical Data Model Recovery
	Design Recovery	Logical Data Model Recovery
Data Structure Conceptualization	Analysis Recovery	Conceptual Data Model Recovery

데이터베이스 리버스 엔지니어링 방법 연구는 크게 3가지로 구분할 수 있다.

첫째, 3NF 나 BCNF를 기본으로 하고 개념 데이터 모델 복원에 필요한 입력(일관된 속성 명, 후보 키, 주 키, 참조 키 등)을 가능한 많이 받는 방법이다[13-14]. 개념

Table. 2 Database Reverse Engineering Methods

Methods	Assumption	Characteristic	Output
Chiang [13]	(i) 3NF or BCNF (ii) consistent naming (iii) Considering only the relevant FD	the generation of inclusion dependencies and the complete justification of the transformations for EER model	EER Model
Batini [14]		The method reveals the drawback of requiring, earlier in the process, semantic input.	EER Model
Petit [15]	Not Mandatory	The only method that includes the normalization of the relational schema.	EER Model
Blaha [16]		This method gives several clues on how/where the user should look for the types of information needed.	OMT model
Alhajj [17]		Generates all possible Candidate and Foreign Keys in an automatic manner	ER Model
Yeh D [18]	Forms represent exact picture of Database schema	Identification of key attributes with least information of keys	EER Model
Mfoura [19]		Generating output by analysis of forms in a human-dependent activity	ER Model

데이터 모델 복원에 필요한 정보가 풍부하여 복원된 개념 데이터 모델의 품질은 좋을 수 있지만, 필요한 입력을 확보하기 위한 노력이 필요하고 적절한 입력이 없다면 적용이 어려울 수 있다.

둘째, 입력의 특별한 전제 없고, 데이터베이스 리버스 엔지니어링을 진행 시 사용 가능한 데이터베이스 정보를 활용하는 방법이다. [15-17] 이들 방법은 적용이 용이하다는 장점이 있지만, 엔지니어의 경험과 능력에 의존성이 큰 단점이 있다. 예외적으로 [17]은 알고리즘을 활용하여 가능한 모든 후보키와 참조키를 자동으로 식별 방법으로 엔지니어의 경험과 능력의 의존이 적다.

셋째, 사용자 화면을 주요 입력으로 개념 데이터 모델을 복원하는 방법으로 [18,19], 화면이 데이터베이스 스키마 형태로 되어 있음을 전제로 한다. 화면에서 사용된 단어와 용어를 활용한다는 것은 사용자가 활용하는 언어로 개념 데이터 모델을 복원할 수 있어 사용자에게 친숙하다는 장점이 있다. 그러나 화면은 애플리케이션 관점의 외부 뷰(External View)로 [20], 복원된 개념 데이터 모델은 특정 화면에만 사용되는 데이터만 표현될 수 있어 범용적으로 사용하기 어려운 데이터 모델이 될 수 있다. 표 2는 연구된 데이터베이스 리버스 엔지니어링 방법을 정리했다.

III. 마스터 데이터를 활용한 데이터베이스 리버스 엔지니어링

본 연구에서 제안하는 데이터베이스 리버스 엔지니어링

의 기본적 접근 방법은 마스터 데이터는 트랜잭션 데이터를 발생시키는 주체가 됨을 전제로 한다. 마스터 데이터는 고객, 상품과 같은 핵심 업무 엔티티 유형으로 식별이 용이하고, 재사용 되어야만 한다. 만약 트랜잭션 데이터 일반 속성으로 마스터 데이터 일반 속성이 있다면, 반 정규화된 속성으로 개념 데이터 모델 복원 시에는 트랜잭션 데이터에서 제거되고 마스터 데이터 속성으로 통합되어야 한다.

위와 같은 반 정규화된 트랜잭션 데이터의 속성을 마스터 데이터로 통합하는 단순한 방법을 활용하여 데이터 품질 문제 때문에 개념 데이터 복원에 어려움이 있는 마이크로 서비스 아키텍처에서도 기존 데이터베이스 리버스 엔지니어링 방법보다 더 쉽게 개념 데이터 모델을 복원할 수 있다. 마스터 데이터를 활용한 데이터베이스 리버스 엔지니어링 단계를 도식화하면 그림 1과 같다.

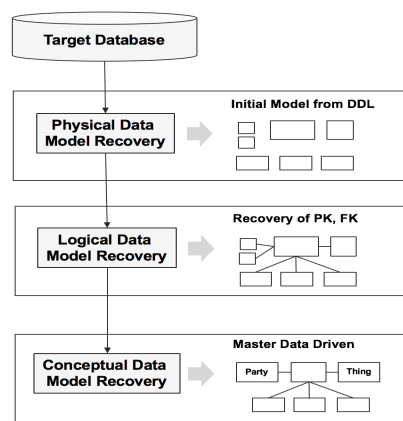


Fig. 1 Reverse Database Engineering Steps Using Master Data

3.1. 물리 데이터 모델 복구

물리 데이터 모델 복구 단계는 DDL(Data Definition Language)를 활용하여 데이터베이스에 구현되어 있는 그대로 초기 데이터 모델을 만든다. DDL을 입력으로 하여 데이터 모델을 만드는 기능은 시장에 나와 있는 대부분의 데이터 모델링 도구에 구현되어 있어, 해당 기능을 활용할 수 있다.

3.2. 논리 모델 복구

논리 데이터 모델 복구 단계는 후보 키와 참조 키를 복원하여 이를 기반으로 초기 모델에 관계와 카디널리티를 추가한다. 참조 무결성 제약이 구현되어 있다면 관계 복원이 용이 하지만 대부분 참조 무결성 제약이 구현되어 있지 않은 경우가 많다. 요즘은 JPA(Java Persistent API)를 활용한 방법으로 데이터베이스에 접근하여 SQL을 직접 구현하지 않고, 애플리케이션에서 SQL이 자동 생성되도록 한다. 그래서 구현된 SQL 확보가 어려워 SQL을 통한 관계 복원[15]도 활용하기 어렵다. 가장 확실한 방법은 데이터값에서 직접 후보 키와 참조 키를 복원하여 관계와 카디널리티를 찾는 방법으로 본 논문에서는 [17]의 데이터 값을 활용한 관계와 카디널리티 복원 알고리즘을 활용한다. 본 논문에서 활용한 알고리즘은 표 3과 같다.

Table. 3 Algorithms for Logical Data Model Recovery

Algorithm	Description
Find Candidate Keys	Find the combination of the unique columns and identify candidate keys.
Find Candidate Foreign Keys	Find the table and column combination that refers to the identified candidate keys.
Find Cardinalities	Find the cardinality of the identified reference keys.

3.3. 개념 모델 복구

개념 데이터 모델 복구 단계는 데이터 의미를 복원하기 위해서 마스터 데이터를 활용한다. 가장 대표적인 마스터 데이터로 Party 와 상품으로 주문, 반품, 교환, 배송, 고객 상담 이력 등의 트랜잭션 데이터 생성에 주체가 된다. 마스터 데이터의 주 키의 특징은 단일 컬럼으로 구성된 대체 키이며, 트랜잭션 데이터의 주 키나 후보 키를 구성하는데 사용된다. 그림 2는 위에 설명한 마스터 데이터와 트랜잭션 데이터 관계를 나타낸다.

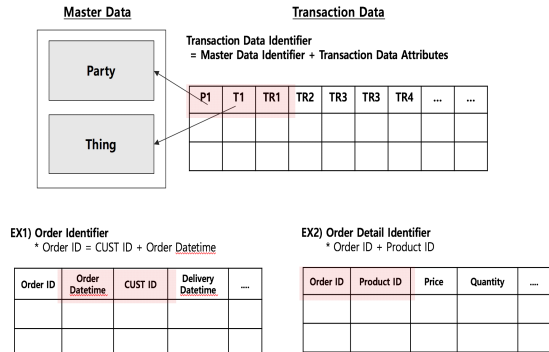


Fig. 2 Master Data and Transaction Data

이러한 마스터 데이터와 트랜잭션 데이터의 관계 특성을 활용하여, 트랜잭션 데이터 일반 속성에 마스터 데이터 속성이 있다면 마스터 데이터로 복원할 수 있다. 이 결과 트랜잭션 데이터에는 트랜잭션에 필요한 데이터 구성 요소만 남게 되어 데이터 구조와 의미 파악이 용이하다.

IV. 적용 사례 및 분석

4.1. 반품 서비스 데이터베이스 적용 사례

본 연구는 마이크로 서비스 아키텍처 기반에서 구현된 데이터베이스를 적용 사례 대상으로 한다. 현재 데이터베이스는 반품 서비스를 위한 데이터베이스로서 데이터 구조와 의미 파악이 어려워 초기 서비스를 구현한 소수 개발자만 유지 보수가 가능하고 반품 리포트 구현이 어렵다는 문제점이 있다. 데이터베이스 리버스 엔지니어링을 수행하기 위한 조사된 입력은 아래와 같다.

- 데이터베이스 구조 : DDL에 한글 설명은 부분적으로 존재하고 참조 키 제약은 없음.
- 데이터 : 데이터 최초 생성은 2014년 6월로, 데이터 분석 시에는 최근 1달로 데이터를 분석함.
- SQL 쿼리 : JPA 기반 개발 방법으로 SQL 쿼리를 확보하기 어려움.
- 화면 및 리포트 : 참고할만한 리포트가 없음.
- 문서 : 데이터베이스 산출물이 존재하지 않음.
- DBMS : MySQL 5.6

활용 가능한 입력인 DDL과 최근 데이터를 활용하여 데이터베이스 리버스 엔지니어링을 수행하였다.

4.1.1. 물리 데이터 모델 복구

대상 테이블은 26개로 주요 특징으로는 1개를 제외한 25개의 테이블의 PK가 자동 순번인 대리(Surrogate) 키로 되어 있다. 물리 데이터모델의 복구 방법은 대상 데이터베이스에서 추출한 DDL을 기반으로 데이터 모델링 툴 (DA# 4.0)을 활용하여 물리 데이터 모델을 복원하였다. 물리 데이터 모델 복구 결과는 APPENDIX 그림 3과 같다.

4.1.2. 논리 데이터 모델 복구

[17]의 데이터 값을 활용한 알고리즘 활용하여 후보 키, 참조 키, 카디널리티를 복원하였고, 3개 주제 영역 (Service A, Service B, Service C)으로 논리 데이터 모델을 구분하였다. 알고리즘 Find Candidate Keys으로 순번으로 구현된 주 키 컬럼을 제외하고 후보 키를 찾는 결과 8개 테이블에서 중복 데이터가 발견되어 0NF이다. 알고리즘 Find Candidate Foreign Keys 으로 후보 참조 키를 찾는 결과 Service A에서 릴레이션쉽 없는 테이블은 2개 발견되었다. 결과는 표4와 같다.

Table. 4 Logical Data Model

Service	Table	Column Count	Normalization
A	A1	27	2NF
	*A2	12	2NF
	A3	8	3NF
	A4	28	2NF
	A5	40	0NF
	A6	8	0NF
	A7	8	0NF
	*A8	8	3NF
	A9	25	0NF
	A10	6	3NF
	A11	9	0NF
	A12	14	2NF
	A13	15	2NF
	A14	19	0NF
B	B1	15	2NF
	B2	6	3NF
	B3	5	3NF
	B4	25	0NF
	B5	9	3NF

Service	Table	Column Count	Normalization
B	B6	8	3NF
	B7	10	3NF
	B8	9	0NF
	B9	10	2NF
C	C1	25	0NF
	C2	7	3NF
	C3	6	3NF

* Tables without Relationships

도출된 1:1 릴레이션쉽은 주제 영역 Service A, Service B에서 각각 발견되었고 통합 고려 대상이다. M:N 릴레이션쉽은 발견되지 않았다. 논리 데이터 모델 복구 결과는 APPENDIX 그림 4와 같다.

4.1.3. 개념 데이터 모델 복구

복구된 논리 데이터 모델 검토 결과 마스터 데이터인 Party와 상품 개념이 컬럼으로만 도출되어 있어, Party와 상품 마스터 데이터를 도출한 후 해당 개념에 속하는 컬럼은 트랜잭션 데이터에서 제거하고 Party와 상품 마스터 데이터에 통합한다. 그 결과 논리 데이터 모델 복원에서 0NF인 테이블은 3NF 테이블로 변경되었고, A4 컬럼 수는 40개에서 25개로, C1 컬럼 수는 25개에서 16개로 줄어들었다.

Table. 5 Conceptual Data Model

Service	Table	Column Count	Normalization	
A	A1	23	3NF	
	A2	9	2NF	
	A4	11	3NF	
	A5	26	3NF	
	A7	8	3NF	
	A8	7	3NF	
	A9	25	2NF	
	A10	6	3NF	
	A14	19	2NF	
	A15	3	3NF	
	A16	3	3NF	
	B	B1	15	2NF
		B4	25	2NF
		B5	8	3NF
		B6	7	3NF
		B8	8	3NF
B9	10	2NF		

Service	Table	Column Count	Normalization
C	C1	16	2NF
	C2	6	3NF
	C3	6	3NF
A,C	* PTY	3	3NF
	* PAD	3	3NF
	* AD	4	3NF
	* PCM	3	3NF
	* CM	3	3NF
A,B,C	* PR	3	3NF
	* VN	1	3NF
	* VNP	4	3NF
	* ITI	3	3NF

* Master Data

그리고 테이블들이 3NF로 변경되는 과정에서 도출된 테이블(A15, A16)도 있고, 통합된 테이블 (A3, A6, A11, A12, A13, B2, B3, B7) 도 있으며, 표 5에 나타냈다. 표5에서 마스터 데이터인 Party 영역은 PTY(Party), PAD(Party 주소), AD(주소), PCM(Party 접촉방법), CM(접촉방법)으로 상품 영역은 PR(상품), VN(업체), VNP(업체상품), ITI(물품식별)으로 표현했다.

또한 Service C 주제 영역은 Service A 주제 영역과 중복이 되는 개념으로 Service A 주제 영역으로 통합할 수 있음을 복구 과정에서 알 수 있었다. 개념 데이터 모델 복구 결과는 APPENDIX 그림 5와 같다.

4.2. 적용 사례 분석

마스터 데이터를 활용한 개념 데이터 모델 복구는 제안 방법으로, 선행 연구의 알고리즘[17]을 사용한 논리 데이터 모델 복구는 기존 방법으로 봤을 때 마스터 데이터를 활용한 결과는 아래 표 6과 같다.

Table. 6 Results of Reverse Database Engineering

	Service					
	A		B		C	
	LDM	CDM	LDM	CDM	LDM	CDM
Average of Normal Form	1.35	2.86	2.11	2.81	2	3
Entity Count	14	20	9	10	3	12
Attribute Count	227	167	91	38	38	55
Relationship Count	11	22	8	11	2	14

* LDM = Logical Data Model, CDM = Conceptual Data Model

마스터 데이터의 활용으로 제안 방법이 기존 방법 보다 평균 정규형(Average of Normal Form)이 향상되었다. 반 정규화된 속성들이 마스터 데이터로 통합되면서 제안 방법이 3 정규형에 가깝게 데이터 모델 품질이 좋아졌다.

엔티티 유형은 마스터 데이터 적용으로 기존 방법이 전체 26개, 제안 방법이 42개로 총 엔티티 유형이 16개 늘었다. 그러나 마스터 데이터인 Party 와 상품 엔티티 타입 9개를 제외하고 트랜잭션 데이터 기준으로는 기존 방법이 전체 26개, 제안 방법이 20개로 감소한 것을 볼 수 있다. 엔티티 유형 수를 마스터 데이터와 트랜잭션 데이터로 나눈 상세 정보는 아래 표 7과 같다.

Table. 7 Detail of Entity type count

Subject Area Data Model		Service A	Service B	Service B
LDM	TD	14	9	3
CDM	MD	Party (5) Product (4)	Product (4)	Party (5) Product (4)
	TD	11	6	3

* LDM = Physical Data Model, CDM = Conceptual Data Model
TD = Transactional Data, MD = Master Data

속성은 기존 방법이 356개 제안 방법이 260개로 줄어든 것을 확인할 수 있는데, 이것 또한 반 정규화된 속성 96개가 마스터 데이터 속성으로 통합되었기 때문이다. 릴레이션HIP 유형은 기존 방법이 21개 제안 방법이 47개로 늘어난 것을 확인할 수 있다.

종합해 보면 마스터 데이터의 활용으로 데이터 모델 품질은 3 정규형에 가깝게 향상됐고, 반 정규화된 속성이 마스터 데이터에 통합되면서 속성수가 줄어들었지만, 엔티티 유형과 릴레이션HIP 유형의 수는 늘어나 개념 데이터 모델의 복잡도는 늘어났다.

V. 결론

마이크로 서비스 아키텍처는 데이터 품질의 희생을 자연스럽게 생각한다. 그 증거로 논리 데이터 복구에서 1NF도 지키지 않은 반 정규화 유형이 발견되었다. 1NF를 지키지 않았다는 것은 개체 무결성을 위반한 것이고, DBMS를 파일 시스템처럼 사용한다는 증거이다.

기존 데이터베이스 리버스 엔지니어링 방법은 최소 1NF가 되어야 적용할 수 있다. 그러나 본 연구에서 제안

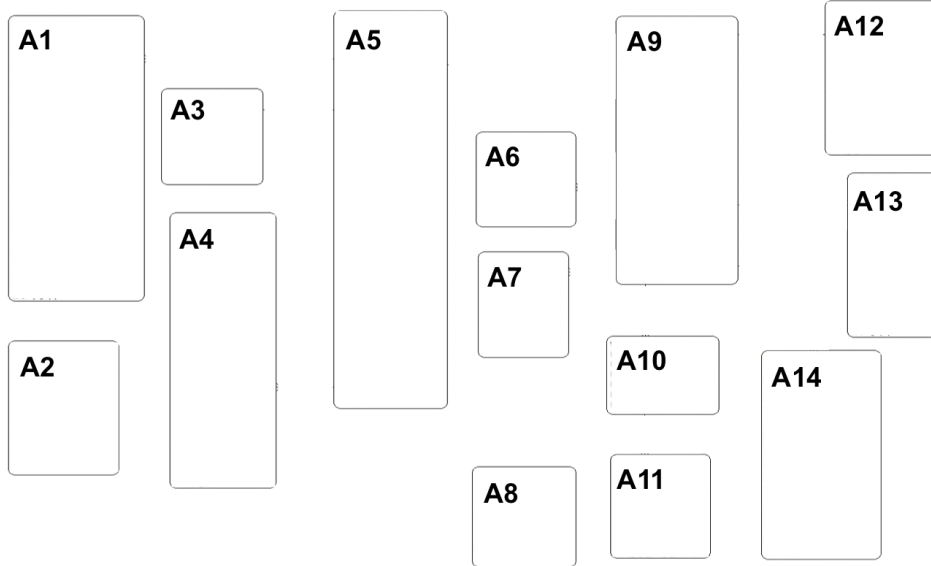
된 마스터 데이터를 활용한 데이터베이스 리버스 엔지니어링 방법은 ONF인 경우에도 적용할 수 있고, 쉽게 식별 가능한 핵심 엔티티인 마스터 데이터를 활용함으로써 실무에서 쉽게 적용할 수 있다는 장점이 있다. 또한 데이터 거버넌스와 데이터 통합에 국한된 마스터 데이터의 활용 영역을 데이터베이스 리버스 엔지니어링 분야로 넓히는데 기여했다. 향후 연구 분야로는 다양한 도메인에 대한 실증적 연구와 함께 복원 개념 데이터 모델을 품질 수준 높이는 방안의 연구이다.

REFERENCES

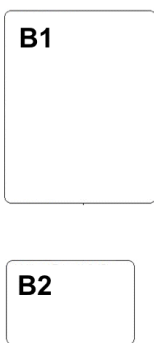
- [1] M. Fowler, and J. Lewis, Microservices a definition of this new architectural term [Internet]. Available: <http://martinfowler.com/articles/microservices.html> , 2014.
- [2] R. Rodger, The Tao of Microservices. *Manning Publications Company*, 2016.
- [3] F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone, "Database design for NoSQL systems," *In International Conference on Conceptual Modeling*, pp. 223-231, Oct. 2014.
- [4] R. Hernandez, Y. Becerra, and J. Torres, "Automatic query driven data modelling in Cassandra," *Procedia Computer Science*, 51, pp. 2822-2826, 2015.
- [5] A. Chebotko, A. Kashlev, and S. Lu, "A big data modeling methodology for Apache Cassandra," *In 2015 IEEE International Congress on Big Data*, pp. 238-245, Jun. 2015.
- [6] K. C. Shin, C. H. Hwang, and H. K. Jung, "Study on NoSQL Data Modeling," *In INTERNATIONAL CONFERENCE ON FUTURE INFORMATION & COMMUNICATION ENGINEERING*, vol. 8, no. 1, pp. 285-287, 2016.
- [7] P. Atzeni, F. Bugiotti, L. Cabibbo, and R. Torlone, "Data modeling in the NoSQL world," *Computer Standards & Interfaces*, 2016.
- [8] D. Pasqualin, G. Souza, E. L. Buratti, E. C. de Almeida, M. D. Del Fabro, and D. Weingaertner, "A case study of the aggregation query model in read-mostly nosql document stores," *In Proceedings of the 20th International Database Engineering & Applications Symposium*, pp. 224-229, Jul. 2016.
- [9] K. Shin, C. Hwang, and H. Jung, "NoSQL database design using UML conceptual data model based on Peter Chen's framework," *International Journal of Applied Engineering Research*, vol. 12, no. 5, pp. 632-636, 2017.
- [10] C. Y. Lee, *Business Ingelligence & Data Analysis*, *Kookmin University Press*, 2016.
- [11] J. L. Hainaut, J. Henrard, D. Roland, V. Englebert, and J. M. Hick, "Structure elicitation in database reverse engineering. In Reverse Engineering," *Proceedings of the Third Working Conference on IEEE*, pp. 131-140, Nov. 1996.
- [12] M. Blaha, "On reverse engineering of vendor databases," *Proceedings Fifth Working Conference on Reverse Engineering*, Honolulu: HI, pp. 183-190, Oct. 1998.
- [13] H. L. Chiang, T. M. Barron, and V. C. Storey, "Reverse engineering of relational databases: Extraction of an EER model from a relational database," *Data & Knowledge Engineering*, vol. 12, no. 2, pp. 107-142, Mar. 1994.
- [14] C. Batini, S. Ceri, S. B. Navathe, "Conceptual Database Design - An Entity-Relationship Approach," *Benjamin/Cummings*, 1992.
- [15] J. M. Petit, J. Kouloumdjian, J. F. Boulicaut, and F. Toumani, "Using Queries to Improve Database Reverse Engineering," *Proceeding ER '94 Proceedings of the 13th International Conference on the Entity-Relationship Approach*, Manchester, pp. 369-386, Dec. 1994.
- [16] W. J. Premerlani, and M. R. Blaha, "An approach for Reverse Engineering of Relational Databases," *Proceedings Working Conference on Reverse Engineering*, Baltimore, pp. 151-160, May. 1993.
- [17] R. Alhaji, "Extracting the extended entity-relationship model from a legacy relational database," *Information Systems*, vol. 28, vol. 6, pp. 597-618, 2003.
- [18] D. Yeh, Y. Li, and W. Chu, "Extracting entity relationship diagram from a table-based legacy database," *Journal of Systems and Software*, vol. 81, no. 5, pp. 746-771, May. 2008.
- [19] N. Mfourga, "Extracting Entity-Relationship Schemas from Relational Databases A Form-driven Approach," *Proceedings of the Fourth Working Conference on Reverse Engineering*, Amsterdam, pp. 184-193, Oct. 1997.
- [20] D. Tsichritzis, and A. Klug, "The ANSI/X3/SPARC DBMS framework report of the study group on database management systems," *Information systems*, vol. 3, no. 3, pp. 173-191, 1978.

APPENDIX

Service A



Service B



Service C

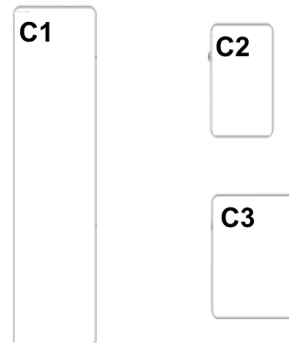


Fig. 3 Physical Data Model

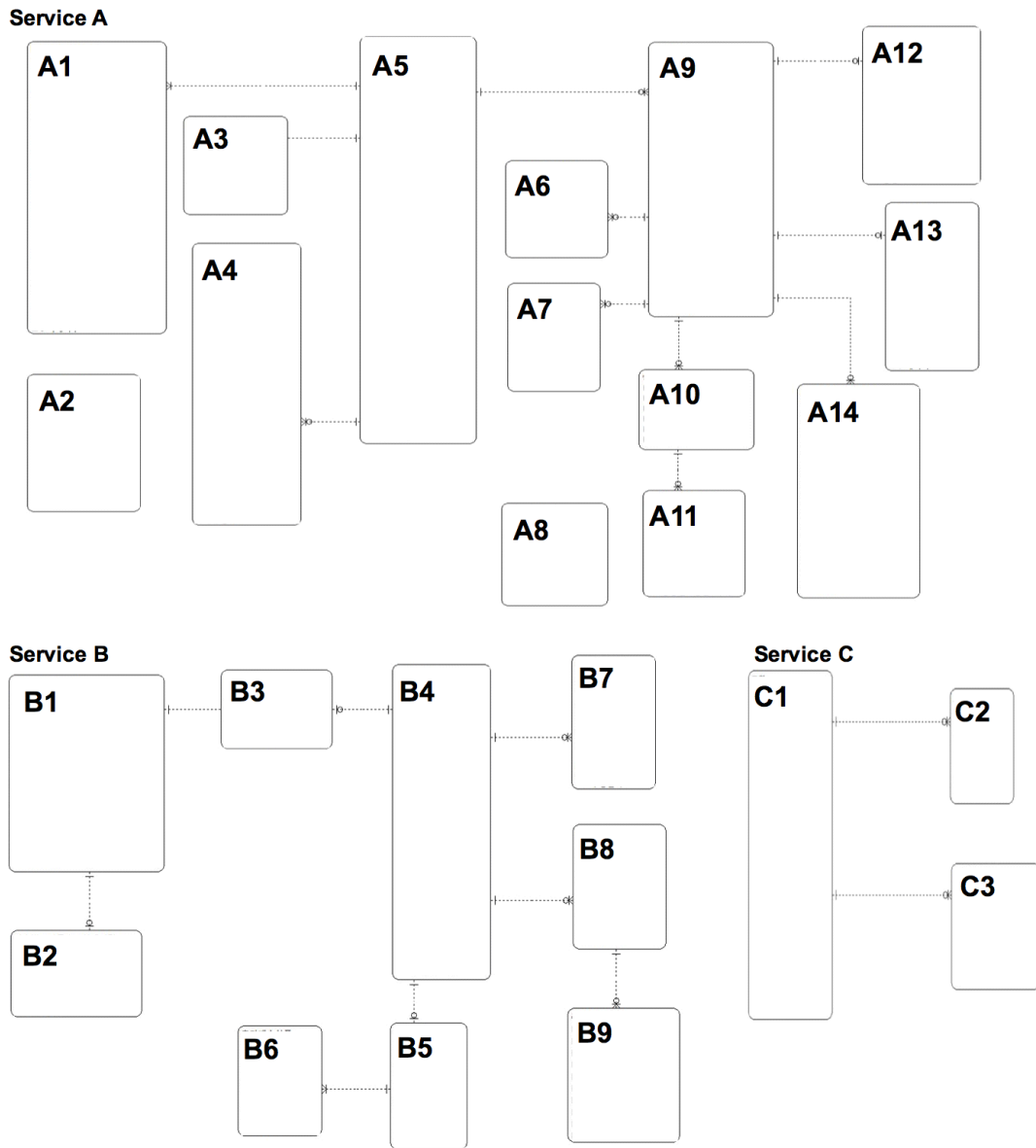


Fig. 4 Logical Data Model

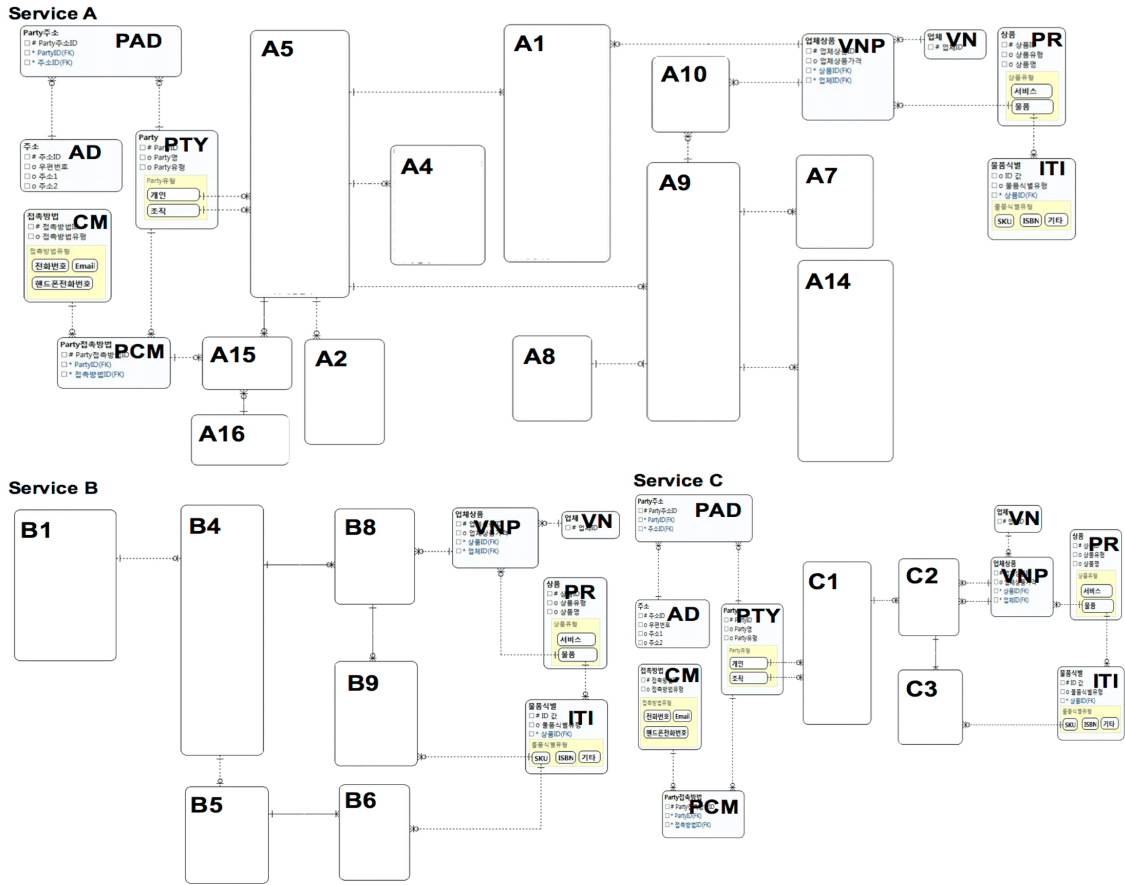


Fig. 5 Conceptual Data Model



신광철(Kwang-Chul Shin)

2007년 2월: 인천대학교 컴퓨터공학 (공학사)
 2013년 2월: 한양대학교 경영전문대학원 (경영학석사)
 2013년 3월 ~ 현재: 국민대학교 비즈니스 IT 전문 대학원 박사 과정
 ※ 관심분야: 리버스 데이터베이스 엔지니어링, 데이터 모델링, 데이터 관리, NoSQL



이춘열(Choon Y. Lee)

1979년 2월: 서울대학교 산업공학과(공학사)
 1983년 2월: 서울대학교 경영학과(경영학석사)
 1990년 2월: 미국 University of Michigan(경영정보학박사)
 1993년 3월 ~ 현재: 국민대학교 경영정보학부 교수
 ※ 관심분야: 데이터 관리, 데이터웨어하우스, 비즈니스 인텔리전스