International Journal of Internet, Broadcasting and Communication Vol.11 No.4 18-25 (2019)
http://dx.doi.org/10.7236/IJIBC.2019.11.4.18

IJIBC 19-4-3

A Data Design for Increasing the Usability of Subway Public Data

Meekyung Min

Dept. of Computer Science, Seokyeong University, Seoul, Korea
mkmin@skuniv.ac.kr

Abstract

The public data portal provides various public data created by the government in the form of files and open APIs. In order to increase the usability of public open data, a variety of information should be provided to users and should be convenient to use for users. This requires the structured data design plan of the public data. In this paper, we propose a data design method to improve the usability of the Seoul subway public data. For the study, we first identify some properties of the current subway public data and then classify the data based on these properties. The properties used as classification criteria are stored properties, derived properties, static properties, and dynamic properties. We also analyze the limitations of current data for each property. Based on this analysis, we classify currently used subway public data into code entities, base entities, and history entities and present the improved design of entities according to this classification. In addition, we propose data retrieval functions to increase the utilization of the data. If the data is designed according to the proposed design of this paper, it will be possible to solve the problem of duplication and inconsistency of the data currently used and to implement more structural data. As a result, it can provide more functions for users, which is the basis for increasing usability of subway public data.

Keywords: *Public Data Portal, Subway Public Data, Data Design, Data Property, File, API*

1. INTRODUCTION

The public data portal is an integrated window for providing public data that is created, acquired, and managed by public institutions in accordance with the government's public data open policy [1]. The data provided by the public data portal includes a variety of areas that are closely related to real life, such as transportation, health, real estate and medical care. Among them, there is a lot of data related to the Seoul Metro. Various policies and studies have been conducted to promote the use of public data [2-5] but there are not many research cases related to public subway data. Related studies include studies on subway boarding and alighting passenger pattern [6], subway congestion prediction [7], and passenger numbers prediction [8]. Public data appears to lack practical applications compared to the enormous amount and variety. Some studies have analyzed the causes of this [9,10]. The lack of utilization is, above all, the inconvenience. In fact, using public data shows that it is very inconvenient to use. The reasons are as follows. First, it is not systematic, such as inconsistent data or lack of up-to-date data. Second, the data retrieval function is not diverse and it is

Manuscript Received: August. 12, 2019 / Revised: August. 18, 2019 / Accepted: August. 23, 2019
Corresponding Author: mkmin@skuniv.ac.kr
Tel: +82-2-940-7750, Fax: +82-2-919-0345
Department of Computer Science, Seokyeong University, Seoul, Korea

inconvenient to use the APIs. This is primarily due to unstructured data. If the data is well defined and the data retrieval capabilities are expanded, the usability is expected to improve.

The data provided by the public data portal comes in two forms: files and APIs. The files are in the format of EXEL, CSV, XML, JSON, etc. Any format can simply be converted to each other and can therefore be regarded as the same format. APIs are usually provided in the form of XML, JSON, or sometimes in the form of SHEET. These data are stored in the database (DB) of public institutions. However, there are many impedance mismatches between the file and the DB. If the fields of the original file are stored in the database as it is, the data is duplicated and not normalized, limiting the search provided to the user. Therefore, ultimately, research on more structured data design is needed to increase the utilization of public data.

In this study, we propose a desirable data design plan by analyzing relevant data of subway public data in Seoul Metro. In the next chapter, subway public data is classified according to its properties. Chapter 3 proposes a design of subway public data and defines the functions that could be provided to users. Chapter 4 concludes the research and gives directions for the future.

2. DATA PROPERTY ANALYSIS

2.1 Static/Dynamic Data

Seoul Metro data can be classified according to various properties, one of which is the static/dynamic properties of the data. Static data is data that is hardly changed after being created once, and dynamic data is data that is constantly changing after being generated.

- **File Data.** The analysis results of the main file data of Seoul Metro according to the static and dynamic properties are shown in Table 1. The data provided in the file form in Table 1 are all static. Some of these are inherently dynamic, but are provided in static form due to the limitations of the inherent characteristics of the file. Some static data have the property of derived data at the same time.

Table 1. Static/Dynamic Properties of Subway File Data

Group	Data	Format	Property
Passenger	Number of passengers (by year, day, hour, station)	File	Static*
Distance & Fare	Distance between stations	File	Static
	Fare by section	File	Static
	Distance and time for transit	File	Static
	Distance and time between stations	File	Static
Station Information	Number of stations in Gu district	File	Static**
	Station address and phone number	File	Static
	Multi language station name	File	Static
Area Information	Major facilities around the station	File	Static
	Bus stops (by station code/external code)	File	Static
Station Facilities	CCTV installation information	File	Static**
	Wifi installation information	File	Static
	Art/Culture pieces	File	Static
	Locker location information	File	Static
	Locker quantity	File	Static
	Official document machine	File	Static
	Entrance and canopy	File	Static

*dynamic property but currently provided in static form

**static and derived

■ **APIs.** Table 2 shows the major APIs of Seoul Metro provided by the public data portal according to their static and dynamic properties. The data provided by the APIs in Table 2 contains static data and dynamic data. Many APIs are provided statically, although they must be provided dynamically. Most APIs are provided in XML and JSON format. SHEET type API can be downloaded as a file and at the same time, it is also shown as SHEET on the screen.

Table 2. Static/Dynamic Properties of Subway APIs

Group	Data	Format	Property
Station & Station Facilities Information	Transit information	API	Static
	Station transit information (by line number)	API	Static
	Nearest station (by coordinates)	API	Static
	Wifi location information	API	Static
	Entrance information	API	Static
	Subway lines information	API	Static
	Subway stations information	API	Static
Train Operation	Train operation schedule information	API	Static
	First/last train information	API	Static*
	Real time train arrival information	API	Dynamic
	Real time train location information	API	Dynamic
	First/last train information (by line no)	SHEET, API	Static*
	First/last train (by code/external code)	API	Static*
	First/last train information (by code/external code)	API	Static*
	Train arrival (by code/external code)	API	Static*
	Train time table (by code/external code)	SHEET, API	Static*
	Train arrival information (transit)	API	Static*
Area Information	Bus stops around the station	API	Static
	Transit information around the station	API	Static
	Major facilities (by code/external code)	API	Static**
	Bus stops (by station code/external code)	API	Static**

*dynamic property but currently provided in static form
**equivalent to file data

2.2 Stored/Derived Data

Stored data is data that must be stored in a database. Derived data refers to data derived from stored data, and need not be stored. A typical example of derived data is statistical data. Statistical data is data derived by calculation from stored data without the need to store itself. Subway public data can also be classified into stored data and derived data according to their properties. As a result of classifying main data according to stored/derived property, Table 3 shows derived data among data of file type. The remaining data not in the table is the stored data.

Table 3. Derived Properties of Subway APIs

Data	Format	Property
Number of passengers (by line, station)	File	Derived
Number of passengers (by line, hour, station)	File	Derived

Number of paid/free passengers (monthly)	File	Derived
Number of alighting passengers (yearly)	File	Derived
Rank, # alighting passengers (yearly)	File	Derived
Number of boarding passengers (yearly)	File	Derived
Rank, # of boarding passengers (yearly)	File	Derived
Number of transit passengers (yearly)	File	Derived
Rank, # of transit passengers (yearly)	File	Derived
Number of passengers (yearly)	File	Derived
Rank, # of passengers (yearly)	File	Derived
Number of tickets (by ticket type)	File	Derived
Number of passengers (by day, station, hour)	File	Derived
Number of transit passengers (by station, data)	File, SHEET, API, chart	Derived

3. DATA DESIGN

3.1 Limitations of the Subway Public Data

This section analyzes the subway data examined in Chapter 2 in terms of its properties and explains its limitations.

The file data and APIs are analyzed in terms of dynamic and static properties as follows. First, since data having dynamic properties is provided in a static form, it cannot provide real time data. Many APIs provide static results, about half of which are originally dynamic data. For example, train schedules, first train information, and last train information are such cases. The train schedule is fixed according to the operation plan, but it is often changed according to the actual operation, so this should be shown in real time. However, since dynamically generated content is not reflected, service of the latest data becomes impossible, which reduces usability. Second, the distinction between file and API is ambiguous. Some APIs are also provided as files. Currently, the provided API only shows the stored file in the form of API, so it cannot reflect real-time data due to the limitation of the file itself.

Next, when looking at the data in terms of stored/derived properties, the problem is that there is little distinction between stored data and derived data. In general, the stored data is stored in the DB, and the derived data is calculated from the stored data when it is needed for data retrieval and then displayed to the user. In the current subway public data, API and file data are ambiguous, so some APIs with derived property are be stored in file or DB with the equivalent fields of the file data.

Also there is a problem that the data cannot be retrieved at various views. Looking at the APIs provided by the current system, it only serves to display the fields in a file's data. There is no API to merge multiple file data or to show the relationship between the file data. This is because the design of the data is not structured. As a result, this causes the failure to provide various retrieval features on the subway data.

3.2 Data Design

This section presents data design proposals for public subway data. Data can be divided into code entities, base entities, and history entities.

- **Code Entity.** The code entity required in the subway database is the entity that manages the code of the subway station. Each subway station has its own code and external code. Currently, the public data portal stores and uses these codes multiple times redundantly in different file data or APIs. For example, there are separate retrievals for the nearby bus stops with the station code and retrieval of

the nearby bus stops with the external code. Most of the file data in Table1 includes all four fields redundantly: station code, station external code, station name, line number, to identify stations. This duplication wastes storage and makes consistent data management difficult. Therefore, separate code entities must be created to manage redundancy to manage station codes. Table 4 below describes the *StationCode* entity for managing the station code. The table lists the attributes and key identifiers for the *StationCode* entity.

Table 4. Code Entity

Code Entity	Attributes	Key Identifier
StationCode	StnID, StnExternalCode, StnCode, StnName, LineNo, EnglishName, CyberStnCode	StnID

- **Base Entity.** The base entity is the basic entity that constitutes a certain database. In the case of subway public data, subway stations are the most basic data. In the subway station entity, the station id representing the station is an identifier of the entity. Examples of the main entities created as a result of the design of the subway data are as follows. Table 5 shows the basic entities related to stations and areas around stations, and the attributes and key identifiers included in each entity. *StnLocation* entity includes location information such as station coordinates and address. *Bus* entity represents the location of the bus stop and *TransToBus* represents the bus information near the station. *StnArea* is information about the business district at the subway exits.

Table 5. Base Entities (Station)

Base entity	Attributes	Key identifier
StnLocation	StnID, Addr, NewAddr, Gu, Phone, XCoord, YCoord, XCoord(WGS), YCoord(WGS)	StnID
TransToBus	StnID, BusStnID	StnID, BusStnID
Bus	BusStnID, BusStnName, XCoord, YCoord	BusStnID
StnArea	StnID, ExitNo, AreaName	StnID

Table 6 shows examples of the base entities in terms of distance, fares and train timetables. *Distance* entity represents the distance between stations. *Fare* entity represents prices based on segment and ticket type. *Timetable* entity is information about a train entering a station, indicating arrival time, departure time, direction, etc., and whether the train is the first or the last train of the day.

Table 6. Base Entities (Station Distance & Fare)

Base entity	Attributes	Key identifier
Distance	StartStnID, EndStnID, Kilometers	StartStnID, EndStnID
Fare	kmSection, TicketType, Price	kmSection, TicketType
TimeTable	StnID, TimeTableID, DayType, UpOrDown, Express, TerminalID, ArrivalTime, DepartureTime, FirstTrain, LastTrain	StnID, TimeTableID

Table 7 below shows examples of base entities associated with station facilities. They are *StnOffice*, *CCTV*, *ArtPieces*, and *Locker*. All of these entities have information about the facilities in subway stations.

Table 7. Base Entities (Subway Facilities)

Base entity	Attributes	Key identifier
StnOffice	OfficeID, OfficeName, OpenDate, Phone, Fax, ZipCode	OfficeID
CCTV	StnID, Location, CoveringArea, Qty	StnID, Location
ArtPieces	StnID, Location, PieceName, Type, Artist, Qty	StnID, Location, PieceName
Locker	StnID, Location, LockerSize, Qty	StnID, Location, LockerSize

- **History Entity.** The history entity is an entity that stores a history of data generated according to time. The train service data is a main example of history entities. A subway train keeps a record every time it runs. Table 8 shows *TrainService* entity, which is an entity of train service data. *TrainService* uses information from the base table, *Timetable*. From this, you can find the first train information, last train information, arrival information, train location information, and so on. When the departure time, arrival time and the actual train ID information are entered, an entity instance is dynamically generated and the history of the subway service information is updated. Another major history entity is the *Passengers* entity, which represents passenger boarding and alighting information.

Table 8. History Entities

History entity	Attributes	Key identifier
TrainService	Date, StnID, TimeTableID, ArrivalTime, DepartureTime, TrainID	Date, StnID, TimeTableID
Passengers	Date, StnID, Hour, NumberOfBoarding, NumberOfAlighting	Date, StnID, Hour

3.3 Retrieval Queries

To retrieve the Seoul Metro information, you must use the APIs by public data portal, or write your own program if there is not the API you want to use in the data portal. Currently, APIs provided by Seoul Metro are inconvenient to use and have limited functions, so in order to improve utilization, various retrieval functions must be provided. The following are the retrieval queries that we can propose based on the assumption that the data is well designed.

. Retrieval of passenger statistics (count, average, rank, etc. according to grouping)
. Retrieval of station list by Gu district
. Subway information retrieval about business district around the station
. Retrieval of distance between subway station and bus station
. Retrieval of rush hour train dispatch interval
. Retrieval of distance and fare between two stations
. Retrieval of a subway station displaying a particular art piece
. Etc.

Also To improve usability, the user interface for queries should be convenient. The UK's railroad public data portal offers users a graphical user interface (GUI) on the screen as well as data files [11]. It includes a number of features for entering search terms, sorting sheets, and selecting conditions. The querying results can be viewed in sheet form or visualized, and you can continue searching again from this screen. Compared to UK data portals, search through the GUI is rarely provided by public data portals in Korea, so this should be improved.

4. CONCLUSIONS

In this paper, we proposed a data design plan for subway public data as a method to increase the utilization of public data of the government. For this purpose, subway public data was classified by several properties. The classification and analysis of the data by static and dynamic properties has shown limitations in the data. Although some data had dynamic properties, there was a problem that the user would only see the static state of the data stored in the file. Some data did not reflect real-time changes. The classification and analysis of the data by stored and derived properties has also shown a problem. Since both stored data and derived data were stored, it was difficult to maintain data consistency. The retrieval feature is so limited that data can retrieved only from fields defined in the file.

Based on the analysis, this study proposed a data design plan to improve the utilization of data. This design classifies entities that can be extracted from subway public data into code entities, base entities, and history entities. Also, a data design plan was proposed to integrate information about subway station, subway operation, passenger information, subway distance and fare.

If data retrieval functions are defined by using the data design plan proposed in this study, and user-friendly APIs and interface are provided, it is expected that the utilization of subway public data can be improved. The future task of this study could be to develop an application system with the database based on the proposed design.

ACKNOWLEDGEMENT

This research was supported by Seokyeong University in 2019.

REFERENCES

[1] B.J. Jeon and H.W. Kim, “An Exploratory Study on the Sharing and Application of Public Open Big Data,” *Information Policy*, Vol. 24, No. 3, pp. 27-41, 2017.
DOI: https://doi.org/10.22693/NIAIP.2017.24.3.027.

[2] B.H. Back and I.K. Ha, “A Method for Selective Storing and Visualization of Public Big Data Using XML Structure,” *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 21, No. 12, pp. 2305-2311, Dec 2017.
DOI: https://doi.org/10.6109/jkiice.2017.21.12.2305.

[3] J.Y. Chang, “An Experimental Evaluation of Box office Revenue Prediction through Social Bigdata Analysis and Machine Learning,” *The Journal of the Institute of Internet, Broadcasting and Communication (JIIBC)*, Vol. 17, No. 3, pp. 167-173, Jun 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.3.167.

[4] M.S. Kang, Y.G. Jung, and D.H. Jang, “A Study on the Search of Optimal Aquaculture farm condition based on Machine Learning,” *The Journal of the Institute of Internet, Broadcasting and Communication (JIIBC)*, Vol. 17, No. 2, pp. 135-140, Apr 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.2.135.

[5] H.J. Seo and S.H. Myeong, "Policy Alternatives for User-oriented Public Data Utilization-Focusing on ICT Managers Perception in Private Sector," *Journal of Korean Association for Regional Information Society*, Vol. 17, No. 3, pp. 61-86, Sep 2014.

[6] M.K. Min, "Classification of Seoul Metro Stations Based on Boarding/Alighting Patterns Using Machine Learning Clustering," *The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC)*, Vol. 18, No. 4, pp. 13-18, Aug 2018.
DOI: https://doi.org//10.7236/JIIBC.2018.18.4.13.

[7] J.S. Kim, "Subway Congestion Prediction and Recommendation System using Big Data Analysis," *Journal of Digital Convergence*, Vol. 14, No. 11, pp. 289-295, Nov 2016.
DOI: https://doi.org/10.14400/JDC.2016.14.11.289.

[8] M.W. Kim, *Predicting Subway Passengers Flows by Spatio-Temporal Modeling*, Master Thesis, Seoul National University, Korea, pp. 4-32, Aug 2017.

[9] M.K. Min, "Modeling and Implementation of Public Open Data in NoSQL Database," *International Journal of Internet, Broadcasting and Communication (IJIBC)*, Vol. 10, No. 3, pp. 51-58, Aug 2018.
DOI: http://dx.doi.org/10.7236/IJIBC.2018.10.3.51.

[10] S.H. Lim, W.J. Jang, and S.M. Lee, "Improving Reuse of Public Transport Information in Open Government," *Basic Research Report*, The Korea Transport Institute, pp. 11-129, Oct 2014.

[11] B.H. Kim, "Recent Trend of Big Data Policy Abroad," *The Korea Contents Association Review*, Vol. 12, No. 1, pp. 38-40, Mar 2014.
DOI: http://doi.org/10.20924/CCTHBL.2014.12.1.038.