An Advanced IoT Data Collection Service for Data-centric Smart Cities

Ryong Lee, Minwoo Park, Sang-HwanLee

Abstract Background/Objectives: ***On behalf of the prevalent IoT, Big Data, and AI technologies, the advance of smart cities are being rapidly accelerated based on data which are made out of numerous sensors and used for deep learning techniques to solve various real-world problems instead of humans. However, it is still a difficult problem to collect and integrate data from diverse sources of different forms due to the heterogeneity and the massive volume of data.***

Methods/Statistical analysis: ***In order to support the complicated work to collect various IoT data from different types of data sources, particularly relieving burdens to develop and manage one-time collectors, we developed an IoT Data Collection Service System, with which users can easily design their own data collectors and control their working statuses to gather and store IoT data. Especially, the proposed system features a simplified workflow from creation to activation of user-defined data collectors.***

Findings: ***In this work, based on an IoT data collection service system for smart cities, we attempted to collect real-time urban sensing data and make them visible on a web-based user interface. The data service requires not only an elaborate procedure to enable users easily to conduct their data collection work, but also hiding the complicated task and overcoming the inherent heterogeneity and complexity of data sources. In particular, it is essential to consider various cases of data collecting scenarios to keep the flexibility and the extendibility of the service.***

Improvements/Applications: ***The expected heterogeneity of IoT data sources can be considerably resolved in our data service enabling users to easily collect data and utilize them for supporting higher-levels of data analyses or application services for smart cities.***

Keywords: IoT Data Collection, Data Service, Smart City, Heterogeneity, Data Integration

I. INTRODUCTION

Due to the increasing demands to take advantages of real-world situational and environmental data for smart cities, IoT data collection becomes an important task requiring much more efficiency and extendibility [1]. So far, in most smart city projects, data are generated, shared, and used in distinctive domains only for each own purposes [2,3]. However, today's real-world problems that are expected to be solved with data requires data sharing and integration to consider much diverse aspects of urban phenomena. For instance, conventional urban traffic monitoring has been done mostly by measuring only a small set of sensors deployed around roads in a city. However, increasing demands on better urban traffic controls by predicting the congestions are requiring much detailed, various and agile data to improve the situations such as daily weather, floating population, and sudden accidents. Therefore, it is strongly necessary to use various data to consider sophisticated urban dynamics [4,5**]**, since much more different types of data related to urban dynamics can give a potential chance to comprehensively understand the complicated urban dynamics and build better plans for constructing better and sustainable smart cities.

In this work, in order to meet various demands on data collection for next-generation Data-centric Smart Cities, we develop an IOT Data Collecting Service featuring the following points.

1) **User-defined Data Collector(uDC):** Data collection work usually involves understanding data sources regarding to the data access protocols, data formats with lots of reluctant tasks to manage data collecting programs. With a web-based user interface, users can easily create and execute a data collector(DC).

2) **A Unified Data Model for Spatio-temporal Integration:** In our data service system, it is intended to provide data for search, integration and analysis services. In order to enable those operations, we transform the most critical two attributes of spatial and temporal information in a unified way.

3) **A Flexible and Extendable System for Increasing Types and Volumes of IoT Big Data:** Basically, we assume that there will be always new data sources having different attributes. In order to cover as many types of data sources as possible, we support three data sharing types; 1) MQTT-based sensor data, 2) Open API based data, and 3) CSV-based data file. To hold the collected data, we adopted a NoSQL database and stored data items into collections. In case of data which cannot be easily stored in the database such as video files, we stored them on a file system directly and managed a link to the file. Therefore, all stored data would keep the original forms simultaneously supporting spatial and temporal integration as mentioned above.

Section 2 will describe requirements and issues about the demands on IoT data collection services for smart cities. In order to meet the complicated

Revised Manuscript Received on May 22, 2019.

Ryong Lee, Research Data Hub Center, KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon, 34141, Korea
Minwoo Park* Corresponding author, Research Data Hub Center, KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon, 34141, Korea
Sang-HwanLee, Research Data Hub Center, KISTI, 245 Daehak-ro, Yuseong-gu, Daejeon, 34141, Korea

requirements, we developed a system helping users to design and activate data collectors. In Section 3, we explain the system architecture to support the creation of user-defined data collectors in detail. Section 4 will explain a case study targeting for collecting a real-time dataset using cars equipped with various sensors. Finally, in Section 5, we conclude this paper with a brief description of future work.

II. IOT DATA COLLECTION AS A SERVICE FOR SMART CITIES

Needless to say, data collection is the first step for any kind of data analyses and applications. Compared to the Web space, IoT data are mostly dependent on real-world space and should reflect on real-world incidents from sensing devices. Furthermore, today's fast growing technical advances in IoT devices and AI require much more data to solve various complicated social issues through data instead of human efforts [6]. Therefore, IoT data collection service will be important more and more as a foundation for smart cities.

In order to support data collection tasks for increasing numbers of data sources with heterogeneous data types, we need to build the data collection systems which can cover practically any type of data sources. In public IoT data sources, we found out that most data sources can be distinguished into three types as follows.

1) **On-line sensor data:** Ideally, this type is the most desirable data source which can capture real-time urban situation by sensor devices usually involving network-based data delivery. In this paper, we do not describe the detailed techniques about sensor and network devices. We only focus on the data that they generate. For sensor data collections, we adopted the trendy MQTT protocol by MOSCA, with which data can be delivered from sensor devices to clients through a data breaker. We think that, on behalf of the protocol, we can easily create the data collection service.

2) **Open API based data:** Obviously, this type is the most common method to share data on the Web. In particular, we also target the public IoT data sources which governmental organizations provide for public benefits on the Web. For instance, in Korea, the government built a web site (http://data.go.kr) to collect and share various data publicly available. In this work, we also support the creation of data collectors to receive data from this type of data sources. Significantly, this type of data sources is relatively formalized and stable to obtain data items.

3) **Legacy data file:** Still, lots of IoT data are being shared as data files organized in their own formats. For instance, surveillance camera based video files have a large volume and are usually maintained as the original video files. If a data source provides such video files, it is necessary to store them into a local file system.

If we simply collect data items from different sources and store them into local storages, it would be merely an incompetent data archive consequently not so useful for data services to help smart cities. As mentioned earlier, we tried to develop our data service to support various applications for smart cities. Therefore, we think that there should be at least several functions such as data search, integration and analysis [7-10]. Obviously, depending on lots of heterogeneous datasets, it is practically difficult to make a universal holder to take all different types of data as a generic format. Instead, we decided to standardize at least two attributes of spatial and temporal information regarding to when and where data were made or related.

III. DEVELOPMENT OF AN IOT DATA COLLECTING SERVICE

In this section, we describe our system architecture and difficulties that we faced in our development.

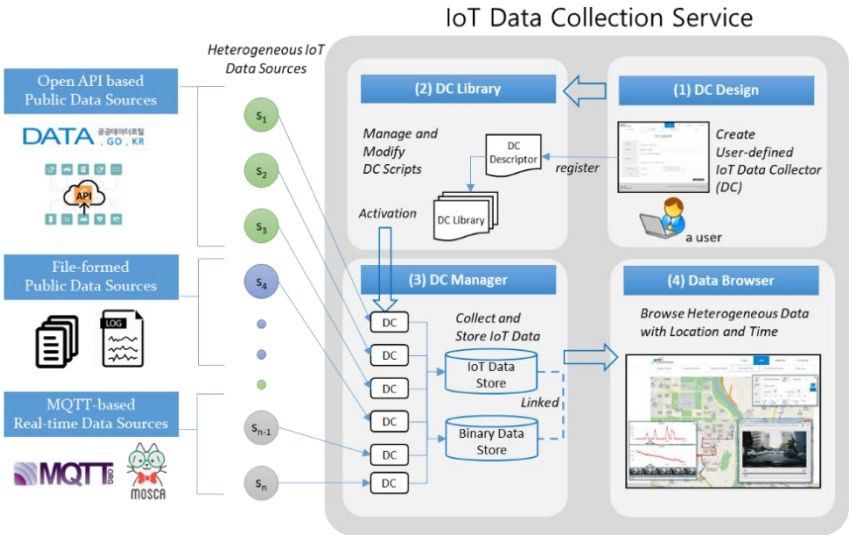

Figure 1. IoT Data Collecting Service System

3.1. Design of an IoT Data Collecting Service

To build an IoT data service, we only assumed that each data source should provide the two attributes of spatial and temporal information on the date items or other descriptions. In our IoT data service, it is consisted of following procedures.

1) **Design of Data Collectors (DC) by users:** at the beginning, a user is supposed to create a data collector for his/her own purpose to access a data source. At the moment, our system provides three types of data sources; i) MQTT-based Sensor Data, ii) Open API based Data and iii) CSV-based data file. Obviously, these three formats would be the most common formats being used for IoT data sharing. In order to create a DC, the system lets user specify the detail of operations which should be made for each data source by a web-based user interface. In the design, it is necessary to provide the basic information about the source such as a user-specified title, URL, access key, ID, password, frequency of fetching, etc. In addition, in order to allow data from each source to be searched and integrated in terms of spatial and temporal aspects, it is possible to set the two fields from the data. Obviously, the two fields are not known in advance and usually have different name characters. Thus, the user needs to choose the two corresponding fields. Furthermore, it is also possible to change the field names if needed, since some fields such as temperature, speed, etc. are frequently required to be represented at fixed terms. At the end of DC design, the system writes out a Descriptor holding the whole information required to actually activate a working DC process.

2) **DC Library and Activation:** All descriptors of DCs are stored and managed in DC library, with which users can call their DC list and perform modification/deletion of DC descriptors. To activate a DC, that is to actually run it, it is necessary to call and execute a descriptor. The activated DC starts to conduct the work following the indications on the descriptor. It first checks the access to the local storage

3) which should be ready in advance. In our system, in order to easily hold the heterogeneous IoT data, we adopted MongoDB which is a NoSQL database taking data without pre-defined data schema, hence simply by pouring data into the database. If the internal conditions for beginning the activation are satisfied, the DC attempts to access the specified data source and fetch the data according to the indications in the descriptor. The data fetching is performed periodically by the specified interval such as hourly, daily, etc. In particular, every data item from each source needs to be examined for finding the spatial and temporal fields and replacing them into local formats for later uses. Data items from each source are stored into distinctive data collections in the database.

4) **Browsing Stored Data**: Finally, all incoming data are stored in the databases respectively into different collections. However, they have at least two common fields, making us search and integrate them for time and space. In order to view the data items stored in the collections, we made a web-based data browser, where each user can perform a conditional data search and browsing. For this, it is necessary to set the search conditions in the fields of target collections, a time period and geographic area on the map interface. On behalf of the indices to the two fields supported by the MongoDB, it is possible easily to search and show the data items which satisfy the conditions on the map interface.

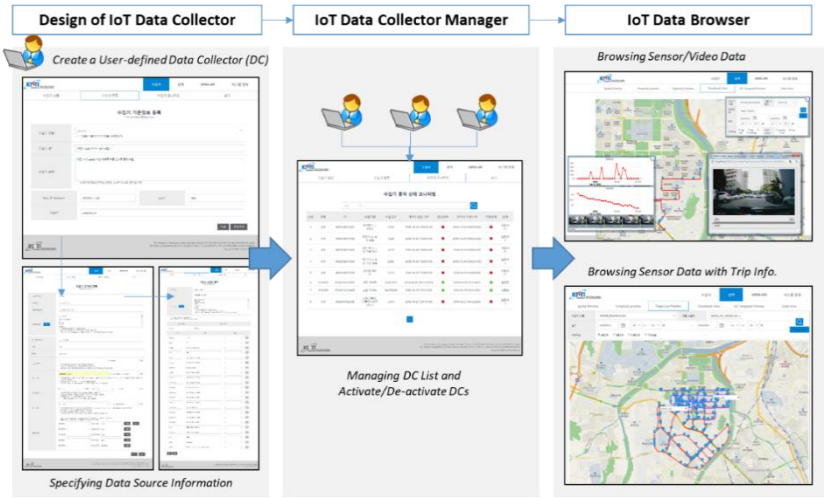

Figure 2. The Workflow from DC Design to Activation

3.2. Heterogeneity of Data Sources

While we attempted to handle as many types of data sources as possible, there are still complicated problems. In order to develop a unified data collecting system, we supposed that all data sources have at least the spatial and temporal information in the data items. However, there are several exceptional cases as follows.

1) **Lack of Spatial Information:** lots of cases

tell that spatial information are not necessary. For instance, a sensor device attached in a fixed location such as a bridge for monitoring the vibration didn't deliver any spatial information, since the location is available on the information page about the source. In practice, due to the reason to save the cost for data delivery on the commercial network such as LTE, such kind of redundant information is often ignored. To deal with such exceptions, we let users add fixed spatial information in the designing step.

2) **Partitioned Information:** Often, information from data sources are partitioned into multiple Open APIs. For example, bus traffic data are separated into bus stops and coded individual bus. In order to exactly know the whole situation, we need to take data from multiple APIs and integrate them according to their own data sharing guidelines. Obviously, this is another type of heterogeneity in terms of data sharing way requiring post data integration.

3) **Linked Video/Image Files**: Video and Image files are also important types of IoT Data. Such data are often provided by additional URLs in the JSON/XML data. In order to obtain such binary data, DCs have to conduct additional tasks by following the links and downloading them. In our architecture, such binary data items are stored in the data file system, organized by directories. Later, it is also necessary to manage the binary data items by keeping the file locations.

In our system, we covered the two types of data sources; MQTT-based sensor data and Open API based data. In addition, there was another type of data files. In fact, lots of IoT are being shared by the form of compressed and organized data files. For such ad-hoc cases, we made our system import such data files from the local storage, with which users need to put raw files into a specified directory. Then, it is necessary to register their metadata detailing their source and schema information. In a case study described in Section 4, we will show such case to import file-based data and how to organize them into the local store making it possible to be searchable and integrated with other datasets in our system.

IV. A CASE STUDY: URBAN DATA COLLECTION AND BROWSING

In order to examine issues of data collection work, we conducted a case study by generating an IoT dataset using sensor-equipped cars for the purpose of urban traffic and environment monitoring. As shown in Figure 3, we attached a sensor box on top of a car, where various sensors are installed such as PM(Particulate Matters), Vibration, VOC, CO, CO2, Temperature, etc. It also estimated the geographic locations by a GPS. Interestingly, it has a dash cam to record the driving scenes which will be used to examine urban traffics from the visual scenes. In particular, the sensor data are stored into a file and sent to a server in our laboratory through the LTE network every second when the sensor box is turn on by the MQTT protocol.

On the other hand, in order to receive and store such real-time data in our system, it is necessary to create and activate a DC in the type of 'Sensor'. The incoming data are then processed according to the indications in the descriptor. In fact, in case of 'Sensor' type, it connects to a MOSCA broker which continuously receive data from the cars. Since the broker works on the topic based data sharing, our sensor-type DC connects to the broker as a client with the topic which should be also specified in the descriptor.

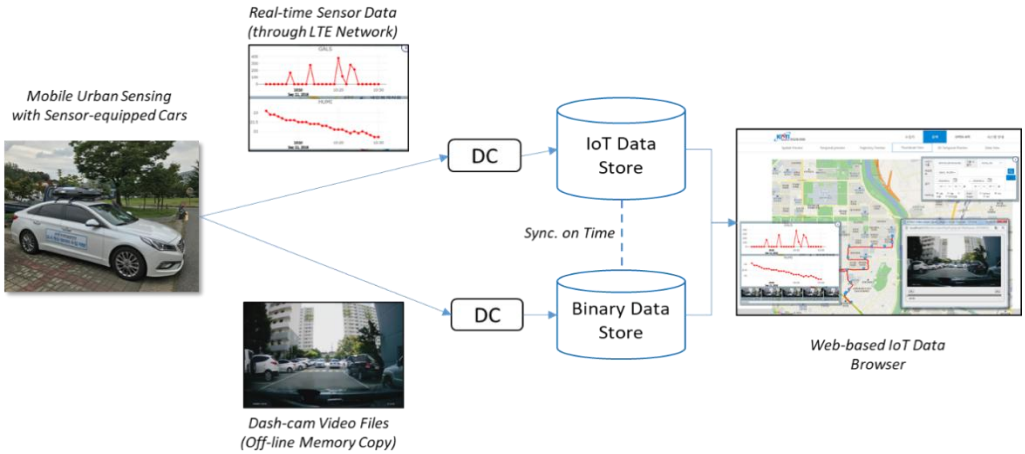

Figure 3. A Case Study for IoT Data Collection using Sensor-equipped Cars

In addition, the video files captured in the cars cannot be transferred to the server through the LTE network to our server directly, since the network capacity is not enough and the cost itself is a limitation. In fact, we ran two cars for 8 hours daily to record video files which reached up to 250GB requiring to be copied directly with the memory to our local store. The recorded video data are separated into 4GB files. For each video file, the beginning time can be known from the file name. Therefore, we added two fields of video file location and their relative time in the video into the corresponding data items which are already stored in the data

collection. That is, the sensor data items are delivered in real time and stored into the data collection in the MongoDB. Later, the video data are imported by the direct file copies using the memory cards due to the large file size. Then, the post-processing for importing video files synchronized with the sensor data by the time reference.

In our system, every data source becomes to have commonality because of the spatial and temporal fields. We only put the minimum condition for the collectors to have such fields. Consequently, we were able to help users easily search, browse and use data in terms of those critical aspects.

Data browsing is practically requiring lots of preparation, since the raw data stored in the data collection usually become a large number. In fact, IoT data collected from smart city environments are monotonically increasing their size. Thus, it is strongly necessary to prepare a data summary for agile data search and browse. Like the conventional data warehouse systems, grid or cluster based data summarization techniques will be very helpful. In our case, we made two approaches. For the trajectory data (time + location), we only selected data items corresponding to every minute experimentally, since we only show the approximated data summarization. On the other hand, non-trajectory data are simply counted with a two-dimensional grid (1km x 1km) for every hour. Likewise, video files are also required to summarized. In our system, we extract video scenes for the time period of samples.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we described our challenge to establish a data collecting service for heterogeneous IoT data sources. It will be very useful to help users collect and integrate different types of data sources. We also explored the usefulness and the limitations of our system applying for a case study of mobile urban data collecting. In our future work, we will study a technique to synchronize various data in terms of space and time. We also experience the difficulties in collecting a large volume of video data who require off-line data collecting and re-arrangement later. Instead of sending a large volume of video data, it is necessary to enable front-end devices to pre-process and extract metadata in advance and only to deliver a small amount of data for improving the performance of data collection.

ACKNOWLEDGMENT

This research was supported by a project 'Establishing a System for Sharing and Disseminating Research Data(K-18-L11-C03)' of Korea Institute of Science and Technology(KISTI), Korea.

REFERENCES

1. Mohsen Marjani, FarizaNasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, Ibrar Yaqoob. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. IEEE Access 5: 5247-5261, 2017.
2. Pavlos Charalampidis, Elias Z. Tragos, Alexandros G. Fragkiadakis. A fog-enabled IoT platform for efficient management and data collection. CAMAD 2017: 1-6.
3. Andreas P. Plageras, Kostas E. Psannis, Christos Stergiou, Haoxiang Wang, B.B. Gupta. Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings. Future Generation Computer Systems, Volume 82, 2018, Pages 349-357.
4. Jun Lee, Kyoung-Sook Kim, Ryong Lee, Sang-Hwan Lee. Visual insight of spatiotemporal IoT-generated contents. AVI 2018: 70:1-70:3.
5. Ryong Lee, Jun Lee, Kyoung-Sook Kim, Minwoo Park, Sang-Hwan Lee. Classification of Urban Districts Based on Road-centric Crowd Movements. In Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Sep. 9-12, 2018, Italy.
6. Federico Montori, Luca Bedogni, Luciano Bononi. On the integration of heterogeneous data sources for the collaborative Internet of Things. RTSI 2016: 1-6.
7. Lawrence A. Klein. Sensor and data fusion: A tool for information assessment and decision making. SPIE Press, 2004, pp. 51. ISBN 0-8194-5435-4.
8. H. F. Durrant-Whyte. Sensor Models and Multisensor Integration. International Journal of Robotics Research, vol. 7(6), pp. 97–113, 1988.
9. Aaron Beach, Mike Gartrell, Xinyu Xing, Richard Han, Qin Lv, Shivakant Mishra, and Karim Seada. Fusing mobile, sensor, and social data to fully enable context-aware computing. In Proc. of the Eleventh Workshop on Mobile Computing Systems & Applications, 2010, pp. 60- 65.
10. Surender Reddy Yerva, Jonnahtan Saltarin, Hoyoung Jeung, Karl Aberer, Social and Sensor Data Fusion in the Cloud. In Proc. of IEEE 13th International Conference on Mobile Data Management, 2012, pp. 276-277.