2019 IEEE International Conference on Consumer Electronics (ICCE)

Construction of Linked Data Platform Implementing Feedback Data Model of Usage Records

Makoto Urakawa, Kenichi Arai, Toru Kobayashi
Nagasaki university, Nagasaki, Japan
bb52218201@ms.nagasaki-u.ac.jp

***Abstract*-- Linked Open Data (LOD) is one of the methods that enables publishing and sharing of data on the Web, and makes data available for many applications. Maintaining the volume and quality of data in LOD requires to keep on engaging people in inputting data. One of the solutions is motivating data producers to input data by giving incentives. However, LOD does not track how data has been used by which applications. In order to track records, data of LOD need to be instantiated to represent who produced it and which applications used it. This paper proposes resource representation by hash value in order to realize instantiation of triple data efficiently according to the concept of RDF which uniquely express instance by URI. This paper also proposes data processing by utilizing the standard SPARQL query to instantiate triple data and feed back usage records to data producers. This paper contributes to showing feasibility of feedback model by building data platform and developing an application enabling people to entry information about spots in town, and by conducting a workshop with people living in Nagasaki city.**

I. Introduction

Linked Open Data (LOD)[1] is one of the methods that enables publishing and sharing of data on the Web. In LOD, data is structured by RDF (Resource Description Framework)[2] and can be described in a format that software can process, hence it is a framework to describe metadata of resources on the Web. By describing the structure and relationship of data on the Web with RDF, it becomes data that many applications can use, which leads to more advanced use of data. Since the Japanese government has been stated to tackle the promotion of opening data owned by local public municipalities. Conversion various data to the open data is actively addressed not only by the administration but also by private companies and citizens. However, motivation for continuous data entry has become an issue. the issue of "the effect, merit, needs of open data is unclear" is highly recognized. That is because LOD does not know how the generated data was used(Fig.1).

In order to encourage people to input data, the crowdsourcing framework is modeled to provides rewards or incentives for participants in return just for work. Data input work in LOD can be furthermore associated with results used from external applications. In this research, we propose a data model and data processing method that can feed back the utilization status of data to the person who input that data. The utilization status represents how many times data is viewed or what applications use data and so on. In order to feedback usage records to producers, it should be identified who generated which data.

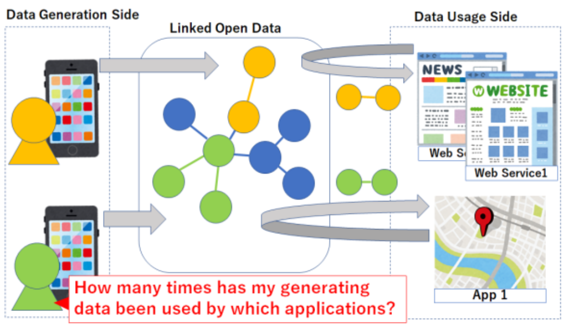

Fig1. Divided aspects of data generation and usage

The scope of this research is expressed in Fig. 2. There are 2 key elements of this scope, one is data model to instantiate each triple data and the other is data processing to make this idea easy to use.

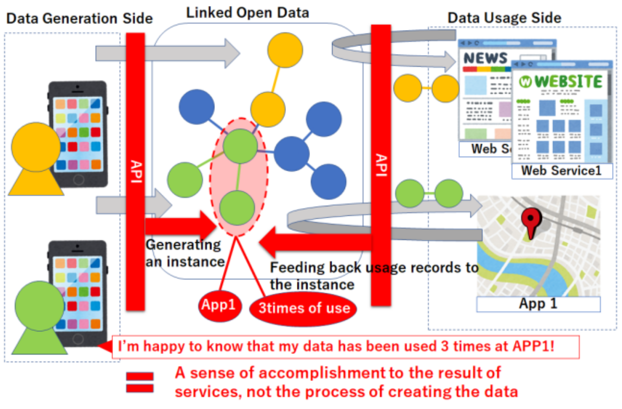
Data Generation Side
Linked Open Data
Data Usage Side
API
API
NEWS
WEBSITE
Web S
Web Service1
Generating an instance
Feeding back usage records to the instance
App1
3times of use
App 1
I'm happy to know that my data has been used 3 times at APP1!
A sense of accomplishment to the result of services, not the process of creating the data

Fig2. The scope of this research

We also developed a Web application and conducted a workshop to enter information of spots in town. This paper concretely shows not only the data structure but also the data processing method to use the data structure. In addition, through the results obtained by the workshop, this research shows the usefulness of the data model.

In this paper, we refers the related research in Chapter 2, and the data structure and data processing are proposed in Chapters 3 and 4. We introduce the platform and Web application in Chapter 5. Then we discuss and conclude in Chapter 6, and Chapter 7.

II. RELATED STUDY

Studies suggesting how it is important to give incentives or rewards to data entry person in order to maintain the volume and quality of data has been done[3] [4] [5]. In addition, ownership of each data is fundamental to engage them[6].

Meanwhile, as a technical study on the data model, several techniques to instantiate triple data has been studied. There are standardized RDF Statement [7], Singleton Properties [8] to identify predicates, Named Graphs [9] which defines multiple triplets at once. RDF Statement creates a new instance and creates a relationship using a specific vocabulary for subjects, predicates, and objects that constitute the triple data. Therefore, the triple number is tripled. Singleton Properties is a method of expressing the predicate of triple data to be instantiated by another ID and using the ID as an instance. In this case, although the required triple number can be suppressed to be doubled, it is necessary to change the predicate which is the important relationship between the subject and the object of the original triple data. Since Named Graphs are devised to group multiple triples, processing to specify the graph is necessary. Singleton Properties is most expressive [10], but original data should be changed, and Named Graphs aims to group multiple triples. Therefore, the data model proposed by this paper is evaluated by comparing it with RDF Statement whose purpose and method are similar.

III. Data Generation Processing

In order to feedback usage records to producers, it should be identified who generated which data, it is necessary to instantiate triple data itself. Fig. 3 shows that triple data about <"latitude (predicate)" of "coin locker (subject)" is "32.74472 (object)"> is expressed as an instance. In this example, information about creator and creating date are also expressed.

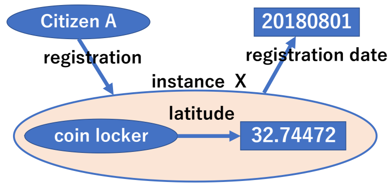

Fig3. Instantiation image of triple data

In this paper, we propose resource representation by hash value in order to realize instantiation of triple data efficiently according to the concept of RDF which uniquely express instance by URI. By concatenating the values obtained by hashing the predicate and the object to the value of subject, it is possible to efficiently maintain uniqueness of the generated instance since the original subject is originally expressed in URI. Fig. 4 shows an example in which Fig. 3 is represented by the data structure proposed by this paper.

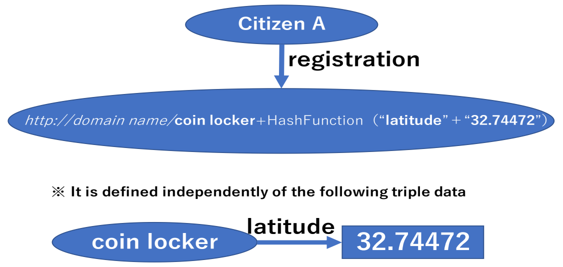
Citizen A
registration
http://domain name/coin locker+HashFunction ("latitude" + "32.74472")
※ It is defined independently of the following triple data
coin locker
latitude
32.74472

Fig4. Proposed instantiation model

As shown in Fig. 4, instantiation is possible without directly affecting the triple data originally generated. Therefore, it is unnecessary to directly provide original triple data with feedback data received from the external application.

A. *Modeling instantiation processing*

A generalized model of the data structure shown in Fig. 4 is shown in Fig. 5. "S", "P", "O" in Fig 5 are values expressing the subject, predicate, object of triple data, and "X" and "Y" are parameters of the subject and predicate that you want to add to triple data.

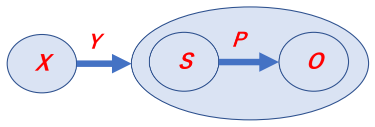

Fig. 5. Data Model for instantiation

Data model for instantiation can be described in the standard SPARQL query shown in Fig. 6. The reason is why calculating hash value is defined in SPARQL.

```
INSERT
{
 ?s ?p ?o.
 ?user ?produce ?spo.
}
WHERE
{
 BIND(URI( S ) as ?s).
 BIND(URI( P ) as ?p).
 BIND(STR( O ) as ?o_original).
 BIND(IF(contains(?o_original,"http"), URI(?o_original),STR(?o_original)) AS ?o)

 BIND(CONCAT(STR(?p),STR(?o)) as ?po).
 BIND(URI(CONCAT(STR(?s),SHA1(?po))) as ?spo).

 BIND(URI( X ) as ?user).
 BIND(URI( Y ) as ?produce).
}
```

Fig6. SPARQL query for instantiation

IV. Data Feed Back Processing

As shown in Chapter 3, since triple data is instantiated, it becomes possible to add information such as registrant information and registration date to each triple data. In this chapter, the data processing for feedbacking usage records such as the number of times of use from the external application will be described in detail.

It is necessary for data producers to manage not only usage records such as the number of times of being displayed or clicked but also information about the application feedbacking them. In addition, it is necessary for each application to flexibly set the information about feedback such as "viewing count" or "clicking count" and so on. Therefore, this paper proposes the data structure which enables differentiate the relationship between triple data and application, and which also make it easy for the application to set usage records by defining it flexibly (Fig. 7).

In this figure, "Relation instance 1" is an instance for uniquely associating "APP01" with the instantiated triple data. Each application can flexibly set the number of times displayed by the application in "Relation Instance". In Fig. 7, "ngsk: relation instance 1 - view count - 13" represents feedback data. As the number of click (click count) in "APP

02” is 142, feedback data can be flexibly defined by each application. “ngsk: APP 01” and “ngsk: APP 02” in the figure are instances for identifying the application.

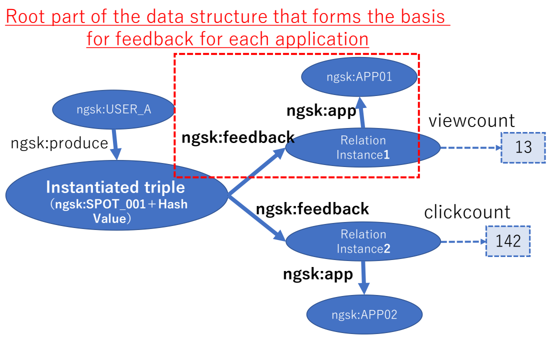

Fig. 7. A data model that enables feedback

A. *Modeling feedback processing*

A generalized model of the data structure shown in Fig. 7 is shown in Fig. 8. "SPO" represents instantiated triple data, and "APP" is identified application, "F" is a parameter for each application to set.

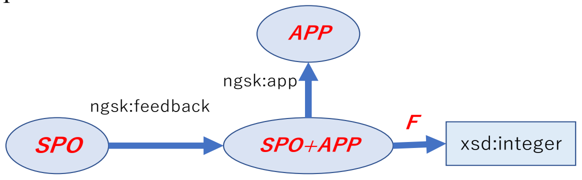

Fig. 8. Data Model for Feed Back

When external applications add their usage records to this data model, they must set the parameters “SPO”, “APP” and “P”. This must be done after accessing LOD data sets, hence they can comprehend a value of “SPO” in advance. “APP” is a parameter to identify themselves and “F” can be determined by themselves. Fig. 9 shows a SPARQL query to update an usage record according to the data mode written in Fig. 8.

```
DELETE {?s <F> ?count.}
INSERT{?s <F> ?countUP. }
WHERE{
< SPO > ngsk:feedback ?s.
?s ngsk:app < APP >.
?s <F> ?count.
BIND(xsd:integer(?count)+1 as ?countUP).
}
```

Fig. 9. SPARQL query Model for Feed Back

Fig. 10 shows the way of feedback usage records to data producers through the data model. This figure combines the data model for instantiation shown in Fig.5 and the data model for feedback shown in Fig.8.

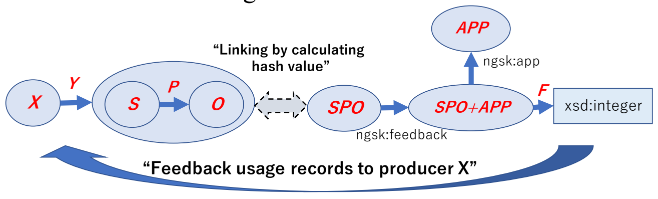
APP
“Linking by calculating hash value”
ngsk:app
X
Y
P
S
O
SPO
SPO+APP
F
xsd:integer
ngsk:feedback
“Feedback usage records to producer X”

Fig. 10. Feedback on the data model

V. LOD PLATFORM

In this section, we introduce the LOD platform constructed to demonstrate that the above data model and generalized query model can be actually implemented in a system. As an application that uses this platform, we developed 2 applications that can register city information related to sightseeing and can refer these city information and add feedback to them.

A. *System Architecture*

From both the security viewpoint and the efficient development, the data processing function and the data storage function are implemented as an API(Fig. 11). The data processing function is configured separately for the data registration application and the data use application. This means that the query statements shown in Fig. 6 and Fig. 9 are installed separately.

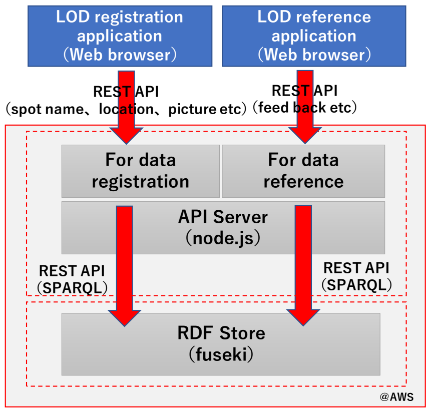

Fig. 11. System Architecture

In the API for data registration, it accepts the spot name, latitude / longitude information and camera image as arguments from the Web application by POST command. At that time, the client application transmits the parameter shown in Fig. 5 as an argument to the API server. On the API server, the SPARQL statement in Fig. 6 is generated and sent to the RDF store in the subsequent stage. Similarly, the data reference API accepts the parameter of Fig. 8 as an argument by the POST command, generates a SPARQL statement, and transmits it to the RDF store.

B. *Prototype of data generation client*

In this section, we introduce the web application using the above data registration API. Fig.12 shows a screen for registering spot information, enabling camera start-up, acquisition of GPS data, etc. from the web browser. The input data can be registered in the RDF store via the registration API. In the data generation client application shown in Fig. 12, there is also a screen for confirming feedback data from the data reference application, which are described in this section D.

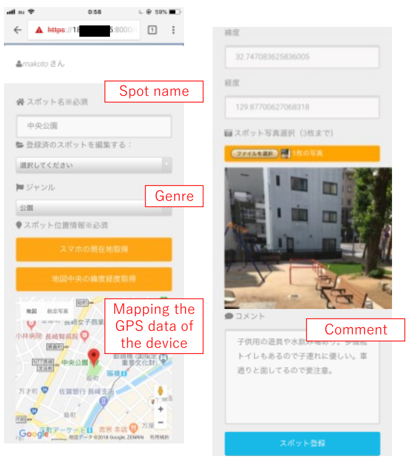

Fig. 12. The screenshot of the registration application

C. *Prototype of data reference application*

As Fig. 13 shows, we also developed a data reference application in order to verify whether feedback about usage records from external application can directly feed triple data. As described above, they can flexibly put feedback data into triple data they accessed. Users of this application can press "Like" and comment on all the spot data. By using the query shown in Fig. 9, the number of pressed "Like" and comments can be registered as feedback data in the RDF store via the data reference API of Fig. 11.

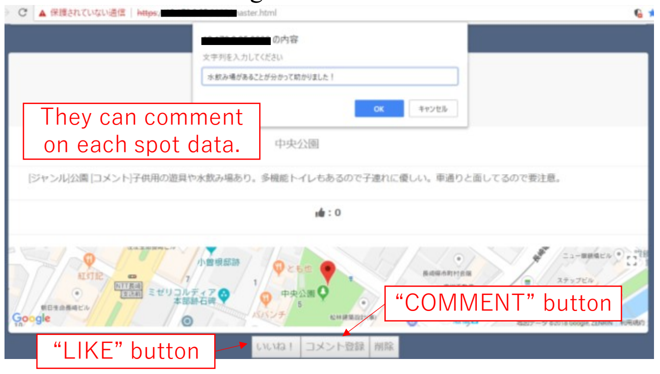

Fig. 13. Example of feedback in reference application

D. *Confirm data feedback in data registration application*

Fig. 14 shows a confirmation screen of spot data registered by a certain entry person in the data registration application. If there is feedback from the external application, you can check its contents. As shown in Fig. 14, you can comprehend the remarks and the number of "Like" pressed in the reference application of Fig. 13.

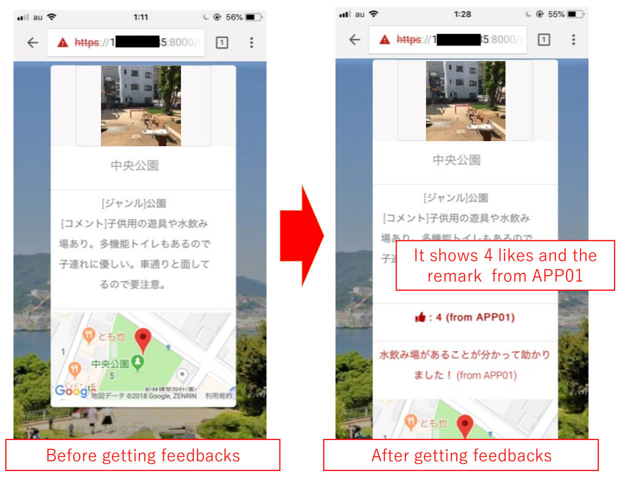
It shows 4 likes and the remark from APP01
Before getting feedbacks
After getting feedbacks
中央公園
: 4 (from APP01)
水飲み場があることが分かって助かりました！ (from APP01)

Fig. 14. Registered spot information confirmation screen

VI. EVALUATION

Using a Web application developed based on the proposed data structure, we held a workshop to have Nagasaki citizens enter city information. In this section, we evaluate the usefulness of feedback data model in LOD from the result of workshop.

A. *Workshop*

Under the cooperation of the Nagasaki Urban Landscape Research Institute[2] , we held a workshop on Nagasaki city on May 27, entering information on streets (parks, toilets, rest house, etc.) while walking around the city. Participants were 7 people living in Nagasaki city, they took 2 hours. As a result, 49 spot data were registered, and the spot list is shown in Fig.15.

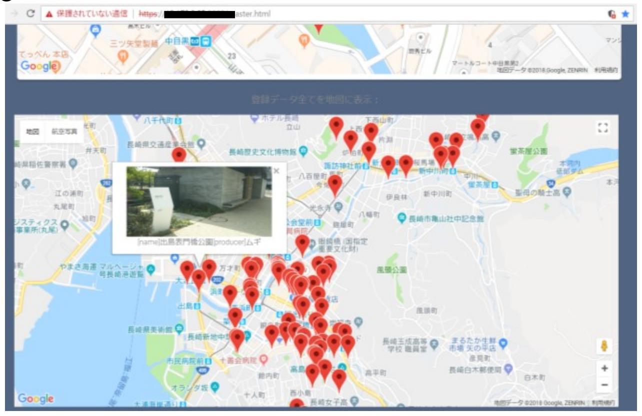

Fig. 15. Registered spot list

We got a comment that access from a web browser made it easy and smooth to register information. In addition, They said that they got a feeling of accomplishment that they can contribute to city planning by receiving feedback. Not only their process but also results or feedbacks from end users encourage them to input data about the town. On the other hand, for example, there was also a problem that different people registered duplicates in the same spot. This problem suggests the need to clarify who and when entered each spot

[2] http://null-project.net/

data, and instantiate it without duplication.

B. *Usefulness of data model*

From the viewpoint of the data model, we evaluate whether it can guarantee who entered when. As shown in Fig. 16, the proposed data model can represent an instance uniquely. In the case of RDF Statement, multiple instantiations (instance X and instance Y in the figure) are possible for the same triple. Therefore, instantiated data can be more easily managed by the data model that this paper proposes. However, multiple participants (Citizen A and B in the figure) can generate triple data representing the fact that they produced in both data models. It has to be limited by applications which use these data models.

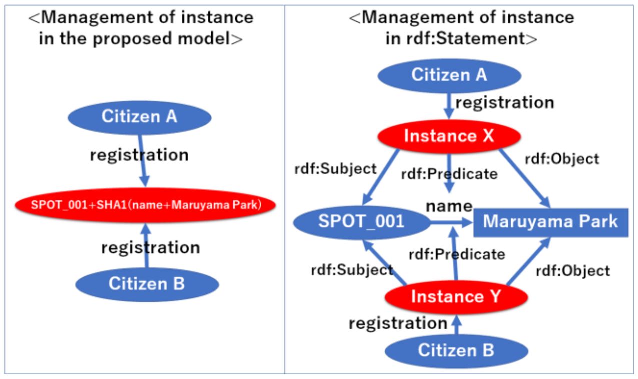

Fig. 16. Duplicate expression in Statement

In the case of rdf:Statement, when instantiating one triple data, one instance and three triples are required. In the proposed data model, since it can be expressed with only one instance, it can be briefly generalized to a query. Furthermore, through application development, we also found that the proposed data model was useful in terms of data management. The reason is that the subject URI of the original triple is adopted and expanded as it is, so that it is structured to make it easier for operators and developers to understand. Also, since it is unnecessary to directly connect with the original triple data to be instantiated, it can be designed as a different database, so it does not affect each other.

VII. CONLUSION

In this paper, in order to realize continuous data generation in LOD, The authors focus on adding the usage records to triple data and feedbacking them to the person who produce the triple data in LOD. In addition, the authors propose the data model and processing in API to achieve LOD platform where records of use can be feedbacked. Furthermore, in order to verify the usefulness of the data model, the authors actually conducted the data entry workshop at Nagasaki city using applications developed for registration and reference. Through the workshop, it was confirmed that the importance of feedback to data producers and the proposed data model are useful.

This paper contributes to showing feasibility of feedback model by verifying the data processing and building data platform and its applications, and conducting the workshop. In the future, we continue to input data by citizens leading to city planning in Nagasaki city, and consider opening API and data store. In this research, we focused on town information on sightseeing, but it is possible to develop widely such as disaster prevention and crime prevention. It is also applicable not only to Nagasaki but also to other cities.

REFERENCE

[1] "Linked Data - Design Issues ". https://www.w3.org/DesignIssues/LinkedData.html.

[2] "RDF 1.1 Semantics". https://www.w3.org/TR/rdf11-mt/.

[3] KARAM Roula, MELCHIORI Michele, "Improving geo-spatial linked data with the wisdom of the crowds," In: Proceedings of the joint EDBT/ICDT 2013 workshops. ACM, 2013. p. 68-74.

[4] TAHARA Yasuyuki, OHSUGA Akihiko, "User Participatory Construction of Open Hazard Data for Preventing Bicycle Accidents," In: Semantic Technology: 7th Joint International Conference, JIST 2017, Proceedings. Springer, 2017. p. 289.

[5] Arakawa Yutaka,Yuki Matsuda, "Gamification mechanism for enhancing a participatory urban sensing," Survey and practical results. Journal of Information Processing, 24(1), 31-38,2016.

[6] SINGH Priyanka, SHADBOLT Nigel, "Linked data in crowdsourcing purposive social network," In: Proceedings of the 22nd International Conference on World Wide Web. ACM, 2013. p. 913-918.

[7] "RDF Schema 1.1 Statement". https://www.w3.org/TR/rdf-schema/#ch_statement.

[8] Nguyen Vinh, Olivier Bodenreider, Amit Sheth, "Don't like RDF reification?: making statements about statements using singleton property," Proceedings of the 23rd international conference on World wide web. ACM, 2014.

[9] Carroll Jeremy J, Bizer Christian. Hayes Pat, Stickler Patrick, "Named graphs," Web Semantics: Science, Services and Agents on the World Wide Web, 3(4), 247-267, 2005.

[10] Hernández Daniel, Aidan Hogan, Markus Krötzsch. "Reifying RDF: What works well with wikidata?." SSWS@ ISWC, 1457, 32-47, 2015.

[11] "RDF 1.1 Query Language".https://www.w3.org/TR/sparql11-query/.