Undefined 0 (0) 1
IOS Press

Applying Linked Open Data for Green Introduction

Takahiro Kawamura [a,*], Akihiko Ohsuga [b]
[a] *Corporate Research & Development Center, Toshiba Corp.*
[b] *Graduate School of Information Systems, University of Electro-Communications, Japan*

Abstract.
The growing environmental consciousness has been causing the concern for urban agriculture and greening business. However, plant cultivation in an urban restricted space is not necessarily a simple matter, and it may extinct depending on species. On the other hand, if it overgrows, it could lead to break the vegetation balance of the surrounding environment. Therefore, we propose an Android application called Green-Thumb Camera, which queries the plants to fit environmental conditions from the LOD cloud based on smartphones sensor information, and then overlays its form in the space using AR to show an image of the mature plant for amateurs. In this paper, after description of LOD content generation method and the application details, we show the evaluation of accuracy of LOD content and usability of the application.

Keywords: Linked Data, AR, Sensor, Green

1. Introduction

Urban greening and agriculture have been receiving increased attention owing to the rise of environmental consciousness and growing interest in macrobiotics. However, the cultivation of greenery in an urban restricted space is not necessarily a simple matter. In particular, as the need to select greenery to fit the space is a challenge for those without gardening expertise, extinction or overgrowth may occur. In regard to both exterior and interior greenery, it is important to achieve an environmental and aesthetic balance between the greenery and the surroundings, but it is difficult for amateurs to imagine the future form of the mature greenery. Even if the user checks images of mature greenery in gardening books, there will inevitably be a gap between the reality and the user's imagination. To solve these problems, the user may engage the services of a professional gardening advisor, but this involves cost and may not be readily available. Therefore, we considered it would be helpful if a mobile service offering the gardening expertise were available on the user's smartphone. In this paper, we propose Green-Thumb Camera, which recommends a plant to fit the user's environmental conditions (sunlight, temperature, etc.) by using a smartphones sensors. Moreover, by displaying its mature form as 3DCG using AR (Augmented Reality) techniques, the user can visually check if the plant matches the user's surroundings. Thus, a user without the gardening expertise is able to introduce a plant to fit the space and achieve aesthetic balance with the surroundings.

The remainder of this paper is organized by mainly two parts: data generation and service development. The data is for the service, and the service is based on the data. Thus the description of either one alone forms only half of the discussion. First, we introduce problems and approaches of our Green-Thumb Camera in section 2, which is followed by LOD (Linked Open Data) generation for the plant and the plant recommendation mechanism. Then, section 3 shows the evaluation on the accuracy of the generated LOD, and the usability of the recommendation service. Finally, section 4 presents related works and section 5 identifies the future issues.

*Corresponding author.

2. Proposal of Green-Thumb Camera

2.1. Problems and Approaches

The Plant recommendation involves at least two problems. One problem concerns plant selection in accordance with several environmental conditions of the planting space. There are more than 300,000 plant species on the Earth, and around 4,000 plant species exist in Japan. Also, their growth conditions involve a number of factors such as sunlight, temperature, humidity, soil (chemical nutrition, physical structure), wind and their chronological changes. Therefore, we have incorporated the essence of precision farming[15], in which those factors are carefully observed and analyzed, and crop yields are maximized through optimized cultivation. In our research, firstly, using the sensors on the smartphone, we determine the environmental factors listed in section 2.3.1, which we consider to be the major factors, and then try to select a plant based on those factors. Other factors, notably watering and fertilizing, are assumed to be sufficient.

Another problem concerns visualization of the future grown form. In addition to achieve the aesthetic balance, overgrowth is an environmental issue. In fact, some kinds of plant cannot be easily exterminated. Typical examples of feral plants are vines such as Sicyos angulatus, which is designated as an invasive alien species in Japan, and Papaver dubium, which has a bright orange flower in spring and is now massively propagating in Tokyo. Therefore, we propose a visualization of the grown form by AR to check it in advance.

Fig. 1 illustrates the service flow of Green-Thumb Camera. First, the user puts an AR marker at the place where he/she wants to grow a new plant, and then taps an Android application, Green-Thumb Camera (GTC), and pushes a start button. If the user looks at the marker through a camera view on the GTC App, the app (1) obtains the environmental factors, such as sunlight, location and temperature from the sensor information (2) searches on LOD Cloud DB with SPARQL, and (3) receives any Plant instances that fit the environment. Then, the app (4) downloads 3DCG data for the plants, if necessary (the data once downloaded is stored in the local SD card), (5) overlays the 3DCG on the marker in the camera view. It also shows two tickers, one for the plant name and description below, and another for the retrieved sensor information on the top. If the user does not like the displayed plant, he/she can check the next possible plant by clicking 'prev' or 'next' button,

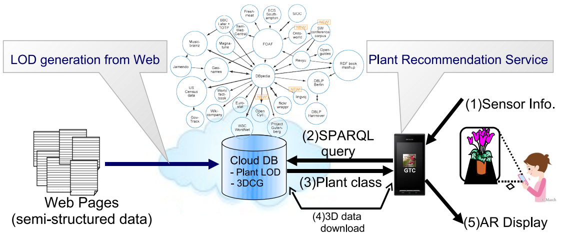

Fig. 1. Service flow of Green-Thumb Camera

or flicking the camera view. Furthermore, if the user clicks a center button, the GTC shows a grown form of the plant (it is the one a year later, but absolutely a calculation).

2.2. Generation of Plant LOD

2.2.1. Plant LOD

First of all, we mention the reason of LOD use for the plant recommendation. As a plant recommendation mechanism, we have already developed three versions. At first, we tried to formulate a function on the basis of multivariate analysis, but gave it up because the setting of priority factors differs depending on the plant species. Next, we tried for automatic adjustment of the setting, and created a decision tree per plant because the reasons for recommendation are relatively easily analyzed from the tree structure. However, this approach obviously poses difficulty for scaling up since training data were manually created. Therefore, as the third one, we build Plant LOD based on collective intelligence on the net, and adopted an approach of selecting a plant by querying with SPARQL. There are already several DBs of plants targeting such fields as genetic analysis and medical applications. However, their diverse usages make it practically impossible to unify those schemas even in future. Furthermore, there are lots of gardening sites for hobbyists on the net, and the practical experience they describe would also be useful. Therefore, instead of a Plant DB with a static schema, we adopted the approach of virtually organizing them on the cloud using LOD. We believe it is one of LOD utilities.

Fig. 2 presents an overview of the generated Plant LOD, in which each plant is an instance of the "Plant" class of DBpedia[2] ontology we referred as a base. DBpedia has already defined 10,000+ plants as types of the Plant class and its subclasses such as "FloweringPlant", "Moss" and "Fern". In addition to that, we created 100 plants mainly for species native to Japan. Each plant of the Plant class has almost 300 Prop-

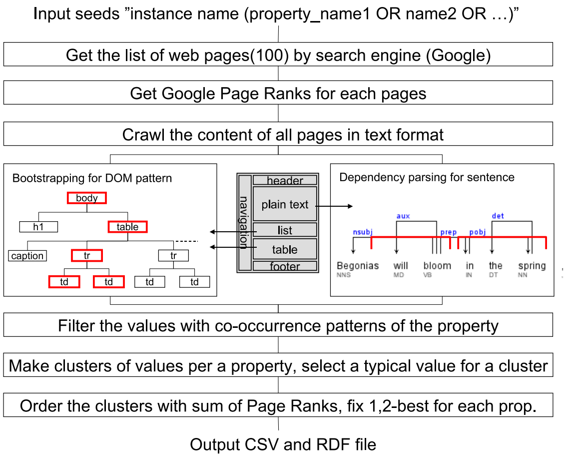

Fig. 3. Process of LOD content generation

erties, but most of them are inherited from "Thing", "Species" and "Eukaryote". So we added 35 properties to represent necessary attributes for plant cultivation, some of which correspond to { Japanese name, English name, country of origin, description, sunlight, temperature (min), temperature (max), planting season (start), planting season (end), blooming season (start), blooming season (end), watering amount, annual grass (true or false), related website, image URL, 3DCG URL, planting area, planting difficulty, etc. }. In Fig. 2, prefixes **gtc:** and **gtcprop:** mean newly created instances and properties. The Plant LOD is written in RDF, and now stored in a cloud DB, DYDRA[1], whose SPARQL endpoint is offered to the public. Openness is not essential for this service, but since it is created from the collective intelligence, and to be expanded by the user, we have made it L'O'D.

2.2.2. LOD generation mechanism

In order to collect the necessary plant information from the Web and correlate it to DBpedia, we developed a semi-automatic mechanism to grow the existing LOD, which involves a bootstrapping method[3] based on ONTOMO[9] and a dependency parsing based on WOM Scouter[8]. However the names of the plant instances can be easily collected from a list on any gardening site and we have already defined the necessary properties from the aspect of the plant cultivation. Therefore, what we need to collect in this case is the value of the property for each plant.

The process of our LOD generation is as follows (Fig. 3). First of all, we make a keyword list, which includes an instance name (plant name) and a logical disjunction of property names such as Rosemary ("Japanese name" OR "English name" OR "country of origin" OR ...), and then search on Google, and receive the result list, which includes more than 100 web pages. Next, we crawl the pages except for pdf files and also check Google PageRank for each page.

As the bootstrapping method, we first extract specific patterns of the DOM tree in a web page based on some keys, which are the property names (and their synonyms), and then we apply these patterns to other web pages to extract the values of the other properties. This method is mainly used for extraction of $(property, value)$ pairs from structured part of a page such as tables and lists.

However, we found there are many (amateur) gardening sites on the Web, most of which are explaining experience of the plant cultivation only in plain text. Therefore, we developed an extraction method using the dependency parsing, because a triple $< plantname, property, value >$ corresponds to $< subject, verb, object >$ in a sentence. It first follows modification relations in a sentence from a seed term, which is the plant name or the property name (and their synonyms), and then extract the above triple, or a triple $< -, property, value >$ in the case of no subject within the sentence ('-' is replaced with the plant name in the keyword list later).

Next, we filter all the property values obtained above if they match to co-occurrence strings prepared for each property name in advance, for example, "temperature" property should obviously co-occurs with °C or °F. Then, we form some clusters of the identical property values for each property name based on LCS (Longest Common Substring). Furthermore, for exclusion of errors, which may be an error of extraction and/or of the information source, we sum up the PageRanks of the source pages for each cluster to determine the best possible property value and the second-best. Finally, after a user selects a correct value from the proposed ones, CSV and RDF files are generated to each plant.

In either way of the bootstrapping and the dependency parsing, the key or seed is retrieved from our predefined schema of Plant LOD, for instance, the instance name and the property name. This is the point to put flesh on the bones of the existing LOD like DBpedia. But, this does not mean to any actual addition to DBpedia. It constructs an another Graph, and is virtually organized with 'sameAs' links.

[1]http://dydra.com/

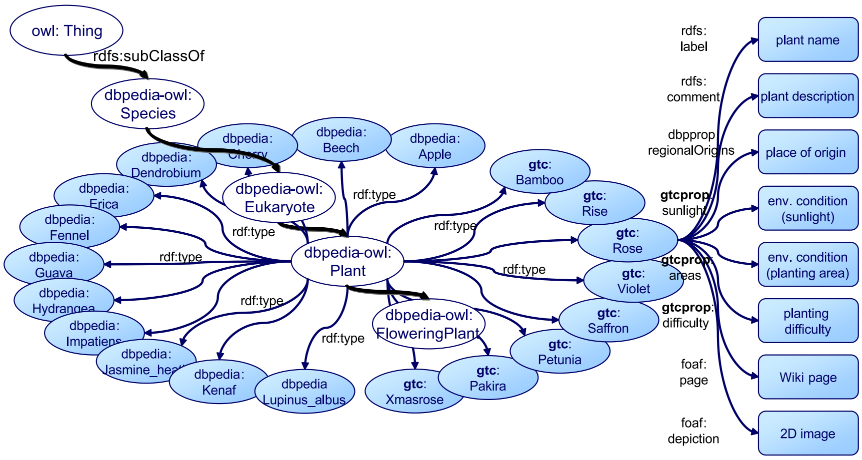

Fig. 2. Overview of Plant LOD

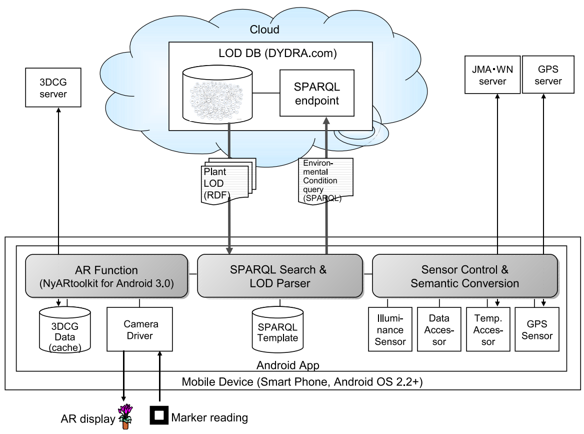

Fig. 4. Architecture of Green-Thumb Camera

2.3. *Development of Recommendation Service*

Fig. 4 shows the architecture of this service. The application requires a smartphone running Google Android OS 2.2+ and equipped with a camera, GPS, and a built-in illuminance sensor. For the AR function, we used NyARToolkit for Android [2] , which is an AR library for the Android OS using a marker. When it detects the predefined marker (Fig. 5) in the camera view, it recognizes its three-dimensional position and attitude, and then displays 3DCGs in Metasequoia format on the marker. The 3DCG can quickly change its size and tilt according to the marker's position and attitude through the camera.

[2]http://sourceforge.jp/projects/nyartoolkit-and/

2.3.1. *Semantic conversion of sensor information*

To begin with we believe semantic search is suited for information retrieval in the field, because input on a smartphone is less convenient, and output of a keyword search is a list of web pages which possibly contain an answer to be found by tapping & scrolling. On the other hand, search with SPARQL, in which the necessary semantic information can be provided, can return the right answer in one shot. Exploitation of mobile and facility sensors is now prevailing, so that the necessary semantic information can be obtained from the sensor. Thus, this scheme of environmental sensing → semantic search → LOD Cloud (← collective intelligence) has great potential of IT support for the field work. This section describes the sensor information we handled, and how we converted the raw data to semantic symbols.

Sunlight

This factor indicates the illuminance suitable for growing each plant and has four levels such as shade, light shade, sunny, and full sun. To determine the current sunlight, we used a built-in illuminance sensor on the smartphone. After the application boots up, if the user pushes the start button at the space where he/she envisages putting a new plant, the illuminance value is measured, and classified to the above levels. If it is less than 300 lux, it is deemed to be a shady area. If it is more than 300 lux but less than 3000 lux, it is deemed to be light shade, and If it is more than 3000 lux but less than 10000 lux, it is deemed to be sunny. Then, if more than 10000 lux, it is deemed to be a full sun area.

Temperature
: This factor indicates the temperature suitable for growing each plant and has the lower and the upper limits of the range. To get the temperature, we referred to past monthly average temperatures for each prefecture from the Japan Meteorological Agency[7] based on the current month and area obtained by the smartphone (described below). But, if the plant is perennial, we check if the every monthly average temperature never exceeds and goes below the range of the plant.

Planting Season
: The planting season means a suitable period (start, end) on a monthly basis for starting to grow a plant (planting or sowing). To get the current month, we simply used the Calendar class provided by the Android OS. However, the season is affected by the geographical location (described below). Therefore, it is set one month later in the south area, and one month earlier in the north area. In the northernmost area, it is set two months earlier, because the periods are given mainly in Tokyo (middle of Japan).

Planting Area
: The planting area means a suitable provincial area for growing a plant. To get the current area, we used the GPS function on the smartphone. Then, we classified the current location (latitude, longitude) for the 47 prefectures in Japan, and determined one of nine provincial areas.

2.3.2. *Plant Recommendation Mechanism*

The GTC App selects a plant based on the above semantic symbols showing the environmental conditions. The SPARQL query includes the environmental conditions in FILTER clause, and is set to return the top three plants out of the instances of the Plant class in the reverse order of the planting difficulty. In the plant cultivation, the state of a farm field and know-how of experts are also important, but a bioscience researcher whom we consulted confirmed that the conditions listed in the previous section are sufficient to serve as the basis for the plant introduction to a considerable extent. In fact, the planting season and area are not independent of the temperature. But, considering these two factors means indirect consideration of other factors like wind, humidity and soil which cannot be easily obtained. We intend to expand the properties of Plant LOD, and to incorporate other environmental conditions in the near future. We are now referring cultivation elements in agroXML[14], which is a standardized language for data exchange in agriculture.

```
SELECT distinct ...
WHERE{
...
FILTER(
...
&&
# Planting Season
( ( (xsd:integer(?start) <= MNT) && (MNT <= xsd:
     integer(?end)) ) ||
  ( (xsd:integer(?start) >= xsd:integer(?end)) &&
    (xsd:integer(?start) <= MNT) && (MNT <= 12) )
        ||
  ( (xsd:integer(?start) >= xsd:integer(?end)) &&
    ( 1 <= MNT) && (MNT <= xsd:integer(?end)) ) )
&&
..
)
ORDER BY ASC (xsd:integer(?difficulty))
LIMIT 3
```

Listing 1: SPARQL query

It should be noted that SPARQL 1.0 does not have a conditional branching statement such as IF-THEN or CASE-WHEN in SQL. Thus, certain restrictions are difficult to express, such as whether the current month is within the planting season or not. Different conditional expressions are required for two cases such as *March* to *July* and *October* to *March*. Although we can express such a restriction using OR and AND in FILTER clause, it is inevitably a redundant expression (see above, where ?start, ?end, and MNT mean the start month, the end month, and the current month respectively). On the other hand, SPARQL 1.1 (W3C Recommendation 21 March 2013) includes IF as Functional Form, so we expect the early dissemination of its implementation.

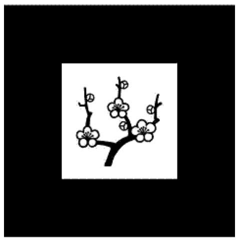

Fig. 5. AR marker (6cm × 6cm)

3. Evaluation of data and service

3.1. *Accuracy of LOD generation*

We applied the LOD generation mechanism to extract the values of the 13 properties for the 90 plants.

Table 1
Extraction accuracy

Accuracy (%)	1-best	2-best	1-best (bootstrap only)	1-best (dependency only)
Precision	85.2	97.4	88.6	85.2
Recall	76.9	87.2	46.2	76.9
Amount Ratio(%)	–	–	10.8	89.2

The result shown in Table 1 includes an average precision and recall of the best possible value (1-best) obtained through the whole process, the bootstrapping method only, and the dependency parsing only, and then those of the second possible value (2-best). It should be noted that we collected 100 web pages for each plant, but some reasons such as DOM parse errors and difference of file types reduced the amount to about 60%. In terms of determining the seasons (start month and end month), if the extracted period is subsumed by the correct period, and the gap between the start (end) months is within 1 month, then it is regarded as correct. Also, in terms of the temperature, if the gap is within 3 $^\circ$C, it is regarded as correct. Properties like description, which are not clear whether it is true or not, are out of scope of this evaluation. If there are more than two clusters whose sum of the PageRanks are the same, we regarded them all as the first position. Also, the accuracy is calculated in units of the cluster instead of each extracted value. That is, in the case of 1-best, a cluster which has the biggest PageRank corresponds to an answer for the property. In the case of 2-best, top two clusters are compared with the correct value, and if either one of the two answers is correct, then it is regarded as correct (thus, it is slightly different than average precision).

$$N - best\, precision = \frac{1}{|D_q|} \sum_{1 \leq k \leq N} r_k$$

,where $|Dq|$ is is the number of correct answers for question q, and r_k is an indicator function equaling 1 if the item at rank k is correct, zero otherwise. The bootstrapping method only and the dependency parsing only mean to form the clusters out of the values extracted only by the bootstrapping and the dependency parsing, respectively. The number of the values in a cluster may vary from more than 10 to 1. Finally, if there are various theories as to the correct value for a property, we selected the most dominant one.

The best possible values (1-best) achieved an average precision of 85% and an average recall of 77%. But, the 2-best achieved an average precision of 97% and an average recall of 87%. So if we are permitted to show a binary choice to the user, it would be possible to present the choice including a correct answer in many cases. The accuracy of the automatic generation would not be 100% after all, and then a human checking is necessary at any step. Therefore, the binary choice would be a realistic option.

In detail, the bootstrapping collects smaller amounts of values (11%), so the recall is substantially lower (46%) than the dependency parsing, but the precision is higher (89%). This is because data written in the tables can be correctly extracted, but lacks diversity of properties. Semantic drift of the values extracted by generic patterns, which is a well-known problem with the bootstrapping method, rarely happened here, because target sources are at most top 100 pages of the Google result, and the values are sorted by the PageRank at the end.

On the other hand, the dependency parsing collects a large amount of values (89%), but it is a mixture of wheat and chaff. But, the total accuracy is affected by the dependency parsing, because the biggest cluster of the PageRank is composed mainly of the values extracted by the dependency parsing. So we are now considering to put some weight on the values extracted by the bootstrapping.

3.2. *Usability of service*

Fig. 6 shows an experiment of the plant recommendation in a rooftop garden. The environment was as follows: Tokyo, December, approx. 5000 lux, 8.4 $^\circ$C. If the user puts the marker in a place where he/she envisages putting a new plant, and sees it through the camera, the GTC App reads the marker and gets the environmental factors such as sunlight, location, and temperature. Then, it overlays 3DCG of a recommended plant on the marker in the camera view. Also, by flicking the camera view, the next plant in the order of recommendation is displayed. In the figure, 3DCG of Wheat (cereal) and/or Rosemary (a medical herb) are displayed as recommended plants. These are typical candidates for planting in this season in Tokyo, and we confirmed the recommendation is working correctly. The GTC App is now open to the public, so anyone can download and try to use it [3].

In order to measure the usability of this service, we conducted two evaluations by a group of potential

[3] http://www.ohsuga.is.uec.ac.jp/~kawamura/gtc.html (in Japanese)

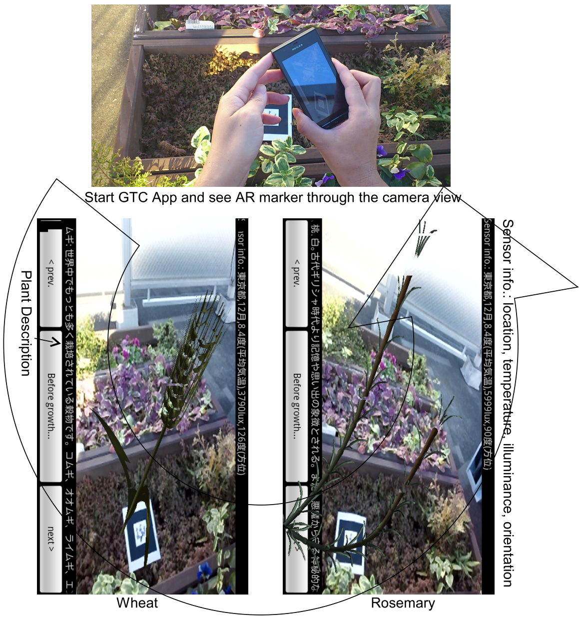

Fig. 6. Experiment of plant recommendation

users who have a liking for the plant cultivation in the home garden. The usability we are referring to here is the definition of ISO9241-11, which says that it is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

At first, we conducted a quantitative evaluation with well-known SD (Semantic Differential) method[11]. In Fig. 7, we identified the highest and the lowest points in the following four metrics defined in ISO9241-11. Effectiveness is the accuracy and completeness with which users achieve specified goals, that is, plant selection. Efficiency is effort expended in relation to the accuracy and completeness with which users achieve the goals. '1' is equal to searching through gardening books. Satisfaction is freedom from discomfort, and positive attitudes towards the use of this service. Context of use is preparation of users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used. As a result, the effectiveness and the efficiency of the plant recommendation which is the main function of this service obtained positive feedbacks even at the lowest point. However, the satisfaction marked a negative point and this reason can be interpreted by the following qualitative analysis.

A second evaluation we conducted is a simplified user testing, which is a way to discover problems on the user interface by carefully observing the users' be-

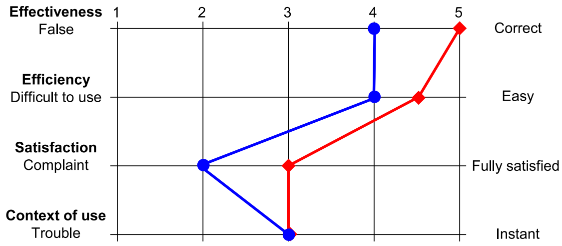

Fig. 7. Result of SD method

haviors and statements through the whole process of task execution. [12] revealed that the user testing with at least five users can discover 85% of the problems. We observed use of the application of five users for 30 hours with interviews, and then confirmed experimental findings, some of which are shown in a list below.

- It was convincing to recommend the plant / vegetables, which are actually growing beside the marker, though it does not offer a new insight.
- Plant information below was sufficient at that time, but it would be better to be click-able to check the price, etc. on the Web later. In addition, the recommended plants should be recorded with its image and location as reference.
- The marker reflects the sunlight in the sunny or full sun, and cannot be easily recognized by the camera. Also, it is occasionally blown away by wind, and contaminated with dirt.
- The more plants for the recommendation, the better. For example, we have just one type of Rosemary so far, but in reality its forms range from upright to trailing.

We will deal with those issues in the future version, especially, as suggested by a comment such that if the marker recognition is quicker it would be really usable, the main reason of the low satisfaction is a failure of the marker recognition. Preparation of the marker like printing and cutting also decreased the point of the context of use. The marker issue is a technical problem, but it always happens that a non-essential problem significantly lower the total usability. Therefore we are now urgently considering a marker-less AR mechanism instead.

4. Related Work

We first introduce researches on LOD content generation. There are at least five ways (and their combinations) to generate LOD. The first one is that an expert writes about a particular theme, e.g. data of Open Government. There is also a way to generate LOD as well as creating Web content using CMS (Content Management Systems). The second and third ones are user participatory creation, e.g. DBpedia and Freebase , and crowdsourcing, e.g. use of Amazon Mechanical Turk. Both of them use the power of the masses, but are classified according to the presence of the business contract. The fourth one is the conversion of the existing structured data like table, CSV and RDB using XLWrap[10] and OntoAccess[6], e.g. Life Science data, and then the last way we think is the (semi-)automatic generation of LOD from the Web. In the recent conferences, researches on the (semi-)automatic generation seems small in number, compared to LOD utilization under the premises that large-scale datasets have been provided. But, one of them is NELL (Never-Ending Language Learner) presented by T. Mitchell at AAAI10[4], which is a semantic machine learning system using the existing ontologies, where several learning methods are combined to reduce extraction errors. Our generation method has been greatly inspired by NELL. However, NELL is targeting the world, so the instances are rich, but the granularity and the number of the properties for each instance is big and limited. On the other hand, by restricting the domain of interest, it is possible for our mechanism to keep the variety and the extraction accuracy of the properties.

Regarding semantic technologies in the agricultural domain, other than DBpedia/Plant and agroXML, FAO (Food and Agriculture Organization of United Nations)[1] is now developing Agricultural Ontology Service Concept Server, whose purpose is the conversion of the current AGROVOC thesaurus to OWL ontologies. AGROVOC is a vocabulary that contains 40000 concepts in 22 languages covering subject fields in agriculture. It is expressed in W3C Simple Knowledge Organization System (SKOS) and also published as LOD. To the best of our knowledge, however, AGROVOC does not include the knowledge for the plant cultivation, and a service like the GTC App has not been offered based on it so far.

In addition, we introduce two researches with respect to combination of sensors and semantics. In Semantic Sensor Network researches, sensor data are annotated with semantic metadata mainly to support environmental monitoring and decision-making. SemSorGrid4Env[5] is applying it to flood emergency response planning. Our architecture is similar to SSN, but instead of searching and reasoning within the collected semantic sensor data, we assume the existence of LOD on the net, to which the sensor data is connected. In that sense, SENSEI[13] had almost the same purpose to integrate the physical with the digital world. But the project mainly addressed the scalability issue and the definition of services interfaces, and then LOD content was limited to a few types of data like geospatial.

5. Conclusion and Future Work

In this paper, we proposed a mobile service, Green-Thumb Camera to enable the users who lack the gardening and agricultural expertise to introduce a plant fitting the environmental conditions. In the near future, we would like to use this service in commercial activity. As described in the beginning, we are now considering the provision of support for greening business which addresses CO2 absorption, and for agribusiness in regard to the food problem. Although we still remain some improvements like the marker-less AR and use of cameras (analysis of the photo of a leaf, for example, enables us to estimate protein content of the plant), this service would be appealing as a step for the precision farming without any capital investment.

Lastly, we summarize the advantage of LOD, and their use of this service. First we are applying a characteristic of linked structures (graph) of data items (not documents) to the information search in the field, and then using a characteristic of a flexible combination of the graphs with different schemas to organize the gardening knowledge from several information sources such as DBpedia and the Web. The openness would be suitable for the collective intelligence, and the future expansion by Human Computation. LOD is a format of the graph data and has no killer application, therefore it would promote LOD to show the services taking advantage of LOD not only for government, bibliographic and scientific data, but also for the average web programmers.

References

[1] Agricultural Information Management Standards, `http://aims.fao.org/`.

[2] S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, Z. Ives: "DBpedia: A Nucleus for a Web of Open Data", Proc. of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, 2007.

[3] S. Brin: “Extracting patterns and relations from the world wide web”, Proc. of WebDB Workshop at 6th International Conference on Extended Database Technology, 1998.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr. and T.M. Mitchell: Toward an Architecture for Never-Ending Language Learning, Proc. of Conference on Artificial Intelligence (AAAI), 2010.

[5] A. Gray, et al.: “A Semantically Enabled Service Architecture for Mashups over Streaming and Stored Data”, Proc. of 8th Extended Semantic Web Conference, 2011.

[6] M. Hert, G. Ghezzi, M. Wursch, H. Gall: “How to "Make a Bridge to the New Town" using OntoAccess”, Proc. of 10th International Semantic Web Conference, 2011.

[7] Japan Meteorological Agency, `http://www.jma.go.jp/jma/indexe.html`.

[8] T. Kawamura, S. Nagano, M. Inaba, Y. Mizoguchi: Mobile Service for Reputation Extraction from Weblogs - Public Experiment and Evaluation, Proc. of Twenty-Second Conference on Artificial Intelligence (AAAI), 2007.

[9] T. Kawamura, I. Shin, H. Nakagawa, K. Nakayama, Y. Tahara, A. Ohsuga: ONTOMO: Web Service for Ontology Building - Evaluation of Ontology Recommendation using Named Entity Extraction, Proc. of IADIS International Conference WWW/INTERNET 2010 (ICWI), 2010.

[10] A. Langegger, W. Wos: “XLWrap - Querying and Integrating Arbitrary Spreadsheets with SPARQL”, Proc. of 8th International Semantic Web Conference, 2009.

[11] S. Himmelfarb: “The measurement of attitudes”, Psychology of Attitudes, A.H. Eagly & S. Chaiken (Eds.), pp.23-88. 1993.

[12] J. Nielsen, T. Landauer: “A mathematical model of the finding of usability problems”, Proc. of ACM INTERCHI (a joint conference of INTERACT and CHI), 1993.

[13] M. Presser, P.M. Barnaghi, M. Eurich, C. Villalonga: “The SENSEI project”, IEEE Communications Magazine, Vol. 47 Issue 4, 2009.

[14] M. Schmitz, D. Martini, M. Kunisch, H. J. Mosinger: “agroXML - Enabling Standardized, Platform-Independent Internet Data Exchange in Farm Management Information Systems”, Metadata and Semantics, pp. 463-468, 2009.

[15] A. Srinivasan: Handbook of precision agriculture: principles and applications, Routledge, 2006.