

Green-Thumb Camera: LOD Application for Field IT

Takahiro Kawamura^{1,2} and Akihiko Ohsuga²

¹ Research & Development Center, Toshiba Corp.

² Graduate School of Information Systems,
University of Electro-Communications, Japan

Abstract. Home gardens and green interiors have recently been receiving increased attention owing to the rise of environmental consciousness and growing interest in macrobiotics. However, because the cultivation of greenery in a restricted urban space is not necessarily a simple matter, overgrowth or extinction may occur. In regard to both interior and exterior greenery, it is important to achieve an aesthetic balance between the greenery and the surroundings, but it is difficult for amateurs to imagine the future form of the mature greenery. Therefore, we propose an Android application, Green-Thumb Camera, to query a plant from LOD cloud to fit environmental conditions based on sensor information on a smartphone, and overlay its grown form in the space using AR.

Keywords: Sensor, LOD, AR, Plant, Field.

1 Introduction

Home gardens and green interiors have been receiving increased attention owing to the rise of environmental consciousness and growing interest in macrobiotics. However, the cultivation of greenery in a restricted urban 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, overgrowth or extinction may occur. In regard to both interior and exterior greenery, it is important to achieve an 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 an 'agent' service offering gardening expertise were available on the user's mobile device. In this paper, we describe our development of Green-Thumb Camera, which recommends a plant to fit the user's environmental conditions (sunlight, temperature, etc.) by using a smartphone's 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 gardening expertise is able to select a plant to fit the space and achieve aesthetic balance with the surroundings.

The AR in this paper refers to annotation of computational information to suit human perception, in particular, overlapping of 3DCG with real images. This technique's development dates back to the 1990s, but lately it has been attracting growing attention, primarily because of its suitability for recent mobile devices. AR on mobile devices realizes the fusion of reality and computational information everywhere. Research[25] on AR for mobile devices was conducted in the 1990s, but it did not attract public attention because "mobile" computers and sensors were big and hard to carry, and the network was slow.

The remainder of this paper is organized as follows. Section 2 describes our proposed service, focusing on plant recommendation and the AR function. Section 3 reports an experiment, and section 4 outlines related work, mainly on AR-based services. Section 5, the final section, presents conclusions and identifies future issues.

2 Proposal of Plant Recommendation Service

2.1 Problems and Approaches

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[1], 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 Table 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. We intend to incorporate other factors in the near future ¹.

Another problem concerns visualization of the future grown form. As well as the need to achieve aesthetic balance for both interior and exterior greenery, overgrowth is an 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 and is now massively propagating in Tokyo. Therefore, we propose visualization of the grown form by AR to check it in advance.

¹ A bioscience researcher whom we consulted confirmed that the factors listed in Table 1 are sufficient to serve as the basis for plant recommendation to a considerable extent.

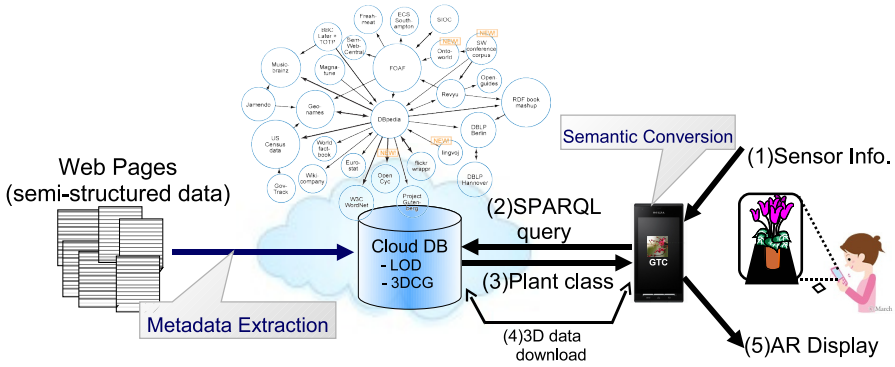


Fig. 1. Service workflow

2.2 Plant Recommendation Service

This section explains the service that we propose.

Service Flow of Plant Recommendation. Firstly, the user puts an AR marker (described later) at the place where he/she wants to grow a 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 (Fig. 1), the app (1) obtains the environmental factors such as sunlight, location and temperature from the sensor information, (2) searches on LOD[2,3] Cloud DB with SPARQL, and (3) receives some Plant classes 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. Fig. 2 is an example displaying “Basil”, a herb. If the user does not like the displayed plant, he/she can check the next possible plant by clicking a left or right button, or flicking the camera view. Furthermore, if the user clicks a center button, GTC shows a grown form of the plant (Fig. 2 below). In this way, the user would be able to find a plant that fits the environmental conditions and blends in with the surroundings. Fig. 3 shows the overview of this service.

Semantic Conversion from Sensor Information to Environmental Factors. This section describes the environmental factors, and how we convert raw data of the sensors to them. Table 1 shows the factors considered in this paper.



Fig. 2. Example of plant display (top: before growth, bottom: after growth)

Table 1. Environmental factors

Factor	Description
Sunlight	minimum and maximum illuminance
Temperature	minimum and maximum temperature
Planting Season	optimum period of planting
Planting Area	possible area of planting

Sunlight

This factor indicates the illuminance suitable for growing each plant and has several levels such as shade, light shade, sunny[4,5].

To determine the current sunlight, we used a built-in illuminance sensor on the smartphone. After the application boots up, if the user brings the

smartphone to the space where he/she envisages putting the plant and pushes the start button on the screen, the sunlight at the space is measured. If it is less than 3000 lux, it is deemed to be a shade area. If it is more than 3000 lux but less than 10000 lux, it is deemed to be light shade, and if more than 10000 lux, it is deemed to be a sunny area. In the case that the sunlight taken by the sensor fits that for the plant, it is deemed suitable.

Temperature

This factor indicates the range (min, max) of suitable temperature for a plant. The lower and the upper limits of the range are determined by reference to the sites as well as to the sunlight.

To get the temperature, we referred to past monthly average temperatures for each prefecture from the Japan Meteorological Agency(JMA)[6], using the current month and area (described below), instead of the current temperature. The temperature for indoor plants from November to February is the average winter indoor temperature for each prefecture from WEATHERNEWS INC.(WN)[7]. In the case that the temperature taken by the sensor is within the range of the plant, it is deemed suitable.

Planting Season

The planting season means a suitable period (start, end) for starting to grow a plant (planting or sowing). The periods are set on a monthly basis according to some gardening sites[8,9].

To get the current month, we simply used 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 for Tokyo (middle of Japan) on most websites. If the current month is in the planting season for the plant, it is deemed suitable.

Planting Area

The planting area means a suitable area for growing a plant. It is set by provincial area according to a reference book used by professional gardeners[4]. 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 the provincial area. If the current location is in the area for the plant, it is deemed suitable.

Plant LOD and SPARQL Query. In this section, we describe how a plant is recommended based on the above factors.

As a recommendation mechanism, we firstly tried to formulate a function on the basis of multivariate analysis, but gave it up because priority factors differ depending on the plant. Next, we created a decision tree per plant because the reasons for recommendation are relatively easily analyzed from the tree structure, and then we evaluated the recommendation accuracy[30]. However, this approach obviously poses a difficulty in terms of scaling up since manual creation of training data is costly. Therefore, we prepared Plant LOD based on collective

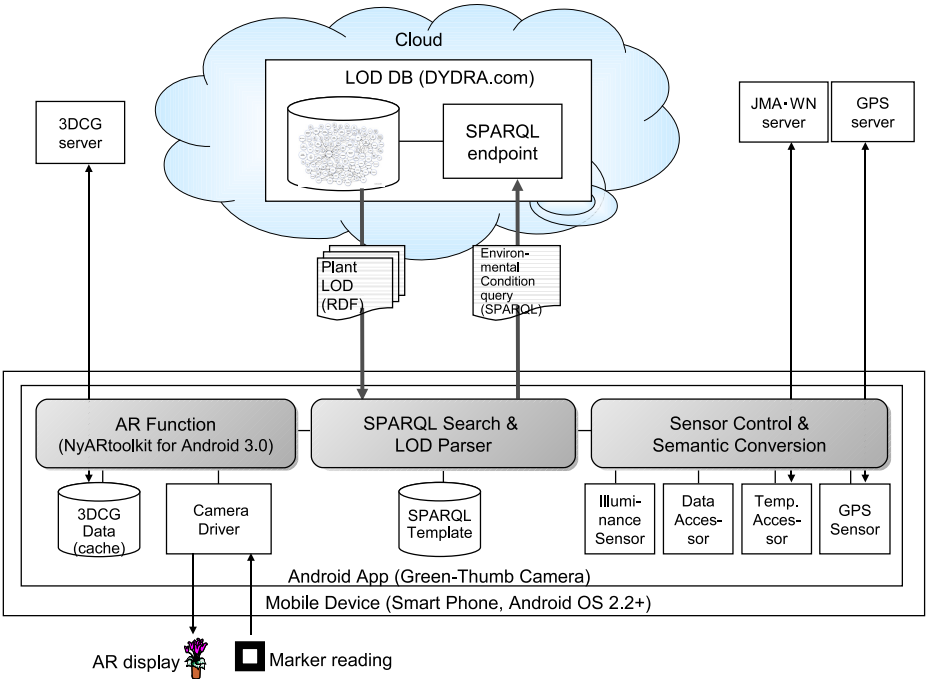


Fig. 3. Service architecture

intelligence on the net and adopted an approach of selecting a plant by querying with SPARQL.

In fact, we consider that SPARQL is intrinsically suited for information search in the field, where a trial-and-error approach to search is difficult because input is less convenient and the network tends to be slower than in the case of desktop search. It is burdensome for users in the field to research something while changing keywords and looking through a list of the results repeatedly. Therefore, search with SPARQL, which can specify the necessary semantics, would be useful in the field.

There are several DBs of plants targeting such fields as gene analysis and medical applications. However, their diverse usages make it practically impossible to unify the schemas. Furthermore, there are lots of gardening sites for hobbyists, 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 using LOD on the cloud. We used a semi-automatic generation system for metadata from web pages that we had developed previously. Then, we created RDF data based on sentence structures and tables that frequently occur on gardening sites, and combined it with the existing Plant LOD (Fig. 1 left).

The Plant LOD is RDF data, in which each plant is an instance of “Plant” class of DBpedia[10] ontology. DBpedia has already defined 10,000+ plants as types of the Plant class and its subclasses such as “FloweringPlant”, “Moss” and

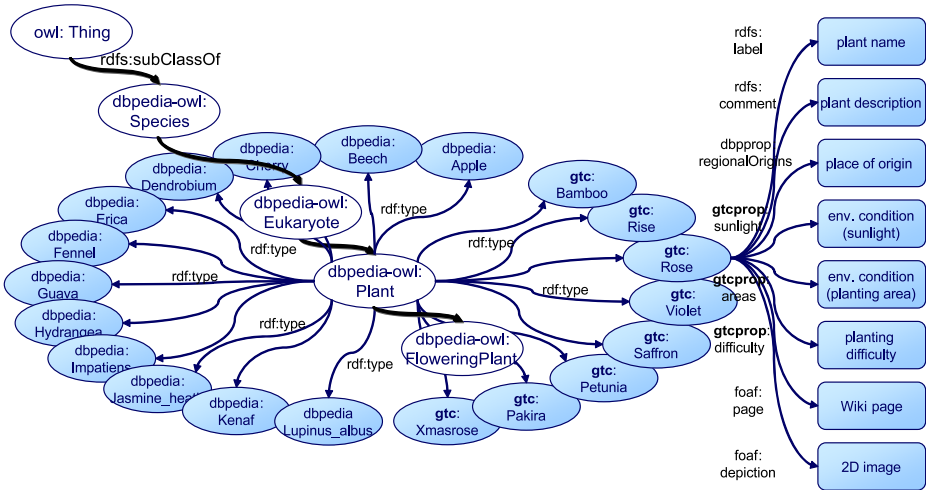


Fig. 4. Overview of Plant LOD

“Fern”. In addition, we created approx. 100 plants mainly for species native to Japan. Each plant of the Plant class has almost 300 Properties, but most of them are inherited from “Thing”, “Species” and “Eukaryote”. So we added 11 Properties to represent necessary attributes for plant cultivation, 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 }. Fig. 4 illustrates the overall architecture of the Plant LOD, where prefixes **gtc:** and **gtcprop:** mean newly created instances and attributes. The Plant LOD is now stored in a cloud DB (DYDRA[11]) and a SPARQL endpoint is offered to the public.

The semi-automatic creation of LOD in this paper is greatly inspired by an invited talk of T. Mitchell at ISWC09[29], and involves a boot strapping method based on ONTOMO[31] and a dependency parsing based on WOM Scouter[32]. But the plant names can be easily collected from a list on any gardening site and we have already defined the necessary attributes based on our service requirements. Therefore, what we would like to collect in this case is the value of the attribute for each plant. As the boot strapping method[12], we first generate specific patterns from web pages based on some keys, which are the names of the attributes, and then we apply the patterns to other web pages to extract the values of the attributes. This method is mainly used for the extraction of $\langle \textit{property}, \textit{value} \rangle$ pairs from structured part of a document such as table and list. However, we found there are many (amateur) gardening sites that explain the nature of the plant only in plain text. Therefore, we created an extraction method using the dependency parsing. It first follows the modification relation in a sentence from a seed term, which is the name of the plant or

the attribute, and then extract triples like $\langle \textit{plantname}, \textit{property}, \textit{value} \rangle$ or $\langle -, \textit{property}, \textit{value} \rangle$. Either way, a key or seed of extraction is retrieved from our predefined schema of Plant LOD to collate the existing LOD like DBpedia. Also, for correction of mistakes, we extracted the values of a plant from more than 100 web sites. If the values are identical, we sum up Google PageRanks of their source sites and determine the best possible value and the second-best. Finally, a user determines a correct value from the proposed ones. We conducted this semi-automatic extraction of the values for the 13 attributes of the 90 plants that we added, and then created the Plant LOD. In a recent experiment, the best possible values achieved an average precision of 85% and an average recall of 77%. We are now conducting more detailed evaluation, thus the results will be discussed in another paper.

The SPARQL query includes the above-mentioned environmental factors obtained from the sensors in FILTER evaluation, and is set to return the top three plants in the reverse order of the planting difficulty within the types of Plant class. 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*. Of course, we can express such a restriction using logical-or(||) and logical-and(&&) in FILTER evaluation, or UNION keyword in WHERE clause. But, it would be a redundant expression in some cases (see below, where ?start, ?end, and MNT mean the start month, the end month, and the current month respectively). On the other hand, SPARQL 1.1 draft[13] includes IF as Functional Forms. So we expect the early fix of 1.1 specification and dissemination of its implementation.

```

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 1.1. SPARQL query

AR Function. This application requires a smartphone running Google Android OS 2.2+ and equipped with a camera, GPS, and an illuminance sensor. For the AR function, we used NyARToolkit for Android[17], which is an AR library for Android OS using a marker. It firstly detects the predefined marker (Fig. 5)

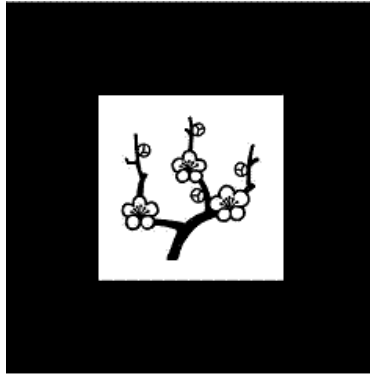


Fig. 5. AR marker (6cm × 6cm)

in the camera view, 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. We have already prepared 90 kinds of plant 3DCG data for recommendation.

2.3 Implementation of LOD/SPARQL for Android

In terms of implementation of SPARQL query and LOD analysis by Java on the Android OS, we developed a library to handle them.

For the creation of SPARQL query, we first prepared some query templates in SPARQL grammar. Then, we select a template as necessary, and replace environmental parameters in the template with the factors obtained from the sensors.

For the analysis of the returned LOD, we first designated the returned format as XML. Some SPARQL endpoints can return the result in JSON (JavaScript Object Notation) format whose grammar is simpler than that of XML. However, parsing a JSON document requires loading of the whole data stream, which consumes the local memory according to the size of the result content as well as XML DOM (Document Object Model) parser. Therefore, we used XML with XmlPullParser of android.util package, which is an event-driven parser like SAX (Simple API for XML), but faster than it.

TAT (Turn Around Time) of query to result is negligible compared to the following procedure to load the 3DCG files. Furthermore, battery consumption poses no problem, unlike in the case of repeating sensor invocations. Note that CPU is Qualcomm Snapdragon 1GHz and connection is WCDMA in our experiment, which is one generation ago.

For more advanced use of LOD/SPARQL in future, however, we are considering the use of ARQoid[16], which is a porting of Jena's ARQ SPARQL query engine to the Android OS, and androjena[15], which is a porting of Jena semantic web framework to the Android OS, although we need to examine the trade-off

of the functionality and performance overhead. Using these libraries, there is an Android Application called Sparql Droid[14]. It can load the local ontology in N3 format stored in the SD card, and query it (or the external SPARQL endpoints) with SPARQL, and it also allows for reasoning over a small ontology.

3 Experimental Test of Plant Recommendation

Fig. 6 shows an experimental result of the plant recommendation. The test environment was as follows: Tokyo, November, 3000+ lux, approx. 10 °C. If the user puts the marker at a place where he/she envisages putting a 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. If the user views the marker from different angles and distances through the camera, it dynamically changes the 3DCG as if it were the real thing. Also, by flicking the camera view, the next plant in the order of recommendation is displayed.

In the figure, 3DCG of a rose and a tulip are displayed as a result. Those are typical candidates for planting in this season in Tokyo, and we confirmed the recommendation is working correctly. To determine it's effectiveness, we would like to conduct some evaluation by a group of potential users in the near future.

4 Related Work

Recently, the remarkable progress of mobile devices has realized the AR function ubiquitously. Mobile devices and AR have a strong affinity because it becomes possible to overlay virtual information on reality everywhere. There are already several reports in the literature and commercial services have been proposed, which can be roughly classified into two categories depending on AR use: to annotate text information to the real object and/or materialize the virtual object in the real scene.

The former includes Sekai Camera[18], which sparked an AR boom in Japan, and Layar[19], VTT(Technical Research Centre of Finland)[20], and Takemura et al.[21]. Sekai Camera displays tags related to the real objects existing in a town, which show the users' comments and reviews. Layar annotates the text information for restaurants, convenience stores and spots on a landscape, and then provides their search function. Research at VTT concerned a system enabling a worker assembling industrial components to see the next parts and how to attach them through a camera. Takemura realized a system employing a wearable computer for annotating information on buildings.

The latter includes My.IKEA[22] and USPS Virtual Box Simulator[23]. My.IKEA realized simulation of furniture arrangement in the user's home through a camera by displaying 3DCG of the furniture on a corresponding marker that comes with a catalog. Virtual Box Simulator is a system to show 3DCG boxes for courier services for determining the suitable box size for an object to be dispatched. The service that we propose in this paper also adopts






CUT	PICTURE	ACTION	DIALOGUE
1		Find a place to put a new plant	“Oh, it has died here!”
2		Put the marker there	
3		See the marker through the camera view	“Show me what you can do!”
4	 <p>Sensor info. 東京,11月,8.4度(平均気温),3279lux,236度(方位)</p> <p>メダカ系が代表的です。花色は赤、桃、黄、白など。また八重咲き、一重咲きがあり、実用品</p> <p>< prev. After growth... next ></p>	(Camera view displaying AR) Flick the camera view to left	“Rose...I don't like it much”
5	 <p>Sensor info. 東京,11月,8.4度(平均気温),3278lux,235度(方位)</p> <p>です。多くの種類があり、園芸上利用されているだけで20種類ほどもあります。花色は赤、</p> <p>< prev. After growth... next ></p>		“Tulip, I like this one”

Fig. 6. Result of plant recommendation

the approach of materializing virtual objects in real scenes and displays 3DCG of non-existent objects as well. However, while the other systems materialize the predefined objects statically bound to the markers, our service materializes more adaptive objects by using the recommendation function according to the real situation estimated by the sensors.

Moreover, we introduce three kinds of research on combining AR with another technique. Regarding the combination with the recommendation function, Guven et al.[24] show 3DCG avatars of real reviewers for a product by reading a marker on the product, and then provide useful information on that product through conversation with the avatars. Our service also shows adaptive information in context with the AR. However, while the AR of this research is only used to show the avatar, AR of our service shows the recommended object itself and overlays it on the real scene to check the aesthetic balance with the surrounding. Therefore, it would be a more practical use of AR.

Regarding the combination with software agents, Nagao et al. proposed agent augmented reality[25] a decade ago and introduced the applications of shopping support and a traveler's guide system.

Furthermore, regarding the combination with plants, there is research by Nishida et al.[26]. They used a 3DCG fairy personifying the plant and whose physical appearance represents the plant's physical condition, thus introducing a game flavor to plant cultivation. Our service also uses AR for the plant growth. However, while they focused on plant cultivation, our service is for the planting and selection of plants and for checking whether they will blend in with the scenery. In fact, there has been little ICT research on plants for non-expert users who enjoy gardening, although precision farming includes agricultural field analysis using sensors for the expert. The most practical service for non-expert users may still be a search engine for the plant names. Focusing on those non-expert users, we provide adaptive information in context by combining the semantic information from the sensor and LOD with AR.

Finally, apart from AR, we introduce two kinds of research regarding sensors and semantics. The first one is Semantic Sensor Network(SSN), in which sensor data is annotated with semantic metadata to support environmental monitoring and decision-making. SemSorGrid4Env[28] is applying it to flood emergency response planning. Our service architecture is similar to SSN. However, instead of searching and reasoning within the mashuped semantic sensor data, we assume the existence of LOD on the net, to which the sensor data is connected.

The second one is about social sensor research, which integrates the existing social networking services and physical-presence awareness like RFID data and twitter with GPS data to encourage users' collaboration and communication. Live Social Semantics(LSS)[27] applied it to some conferences and suggested new interests for the users. It resembles our service architecture in that face-to-face contact events based on RFID are connected to the social information on the net. However, from the difference in its objective, which is a social or field support, the information flow is opposite. In our architecture, the sensor (client) side requests the LOD on the net, although in LSS the social information (DB) collects the sensor data.

5 Conclusion and Future Work

In this paper, we proposed an ‘agent’ service, Green-Thumb Camera, which works on a smartphone equipped with sensors, LOD and AR to enable users who lack gardening expertise to select a plant fitting the environmental conditions.

In the near future, first, we intend to summarize the semi-automatic generation of Plant LOD mentioned in section 2.2. We would also like to apply this framework of environmental sensing → semantic conversion → LOD Cloud (→ AR display) in other fields that would benefit from greater IT support. This vision is expressed in the subtitle of this paper. In particular, we are considering the provision of support for the greening business, which addresses environmental concerns, and for agribusiness in regard to the growing food problem. If the former is the case, target plants would be “Lawn” or “Sedum” in most cases. If the latter is the case, they are “Wheat”, “Rise”, “Corn” and “Bean”. For all those plants, there are sufficient knowledge about their cultivation on the net, but utilization of those knowledge in the field requires laptop PCs and keyword seaches. However, using this framework of the sensor and LOD, we can create a new service for the smartphone, which automatically shows instructions suitable for the current condition of a plant (we did not use a built-in camera to take a photo in this paper, but analysis of the photo of leaf, for example, will enables us to estimate protein content of the plant). It would be appealing as a simplified precision farming without capital investment. In field research, exploitation of mobile and facility sensors is now prevailing, but applications are still vague although sensor information is overflowing. By serving as an intermediary interpreting the semantics of sensor information and connecting it to the collective intelligence on the net, we seek to exploit the tremendous potential of LOD.

References

1. Srinivasan, A.: Handbook of precision agriculture: principles and applications. Routledge (2006)
2. Berners-Lee, T.: Design Issues: Linked Data (2006), <http://www.w3.org/DesignIssues/LinkedData.html>
3. Linked Data - Connect Distributed Data across the Web, <http://linkeddata.org/>
4. Neo-Green Space Design. Seibundo Shinkosha Publishing (1996)
5. engeinavi (in Japanese), <http://www.engeinavi.jp/db/>.
6. Japan Meteorological Agency, <http://www.jma.go.jp/jma/indexe.html>
7. weathernews, <http://weathernews.com/?language=en>
8. Makimo Plant (in Japanese), <http://www.makimo-plant.com/modules/maintenance/index.php>
9. Angyo: The Village of Garden Plants, <http://www.jurian.or.jp/en/index.html>
10. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)

11. DYDRA, <http://dydra.com/>
12. Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: Atzeni, P., Mendelzon, A.O., Mecca, G. (eds.) WebDB 1998. LNCS, vol. 1590, pp. 172–183. Springer, Heidelberg (1999)
13. W3C: SPARQL 1.1 Query Language Working Draft (May 12, 2011), <http://www.w3.org/TR/2011/WD-sparql11-query-20110512/>
14. Sparql Droid, <https://market.android.com/details?id=com.monead.semantic.android.sparql>
15. androjena, <http://code.google.com/p/androjena/>
16. ARQoid, <http://code.google.com/p/androjena/wiki/ARQoid>
17. NyARToolkit for Android, <http://sourceforge.jp/projects/nyartoolkit-and/>
18. Sekai Camera, <http://sekaicamera.com/>
19. Layar, <http://layar.jp/>
20. Woodward, C., et al.: Demonstration of assembly work using augmented reality. In: Proc. of 6th ACM International Conference on Image and Video Retrieval, CIVR (2007)
21. Tenmoku, R., Kanbara, M., Yokoya, N., Takemura, H.: Annotation overlay with a wearable computer using augmented reality. In: Proc. of 1st CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing (2002)
22. My.IKEA, <http://www.youtube.com/watch?v=0javjTvzIMw>
23. USPS: Priority Mail - Virtual Box Simulator, <https://www.prioritymail.com/simulator.asp>
24. Guven, S., Oda, O., Podlaseck, M., Stavropoulos, H., Kolluri, S., Pingali, G.: Social Mobile Augmented Reality for Retail. In: Proc. of IEEE International Conference on Pervasive Computing and Communication, PerCom (2009)
25. Nagao, K.: Agent Augmented Reality: Agents Integrate the Real World with Cyberspace. In: Ishida, T. (ed.) Community Computing: Collaboration over Global Information Networks (1998)
26. Nishida, T., Owada, S.: MOEGI: Plant Fostering by the Assistance of Augmented Reality. In: Proc. of 14th Workshop on Interactive Systems and Software, WISS (2006)
27. Szomszor, M., Cattuto, C., Van den Broeck, W., Barrat, A., Alani, H.: Semantics, Sensors, and the Social Web: The Live Social Semantics Experiments. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 196–210. Springer, Heidelberg (2010)
28. Gray, A.J.G., García-Castro, R., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.-P., Page, K., Sadler, J., Frazer, A., Galpin, I., Fernandes, A.A.A., Paton, N.W., Corcho, O., Koubarakis, M., De Roure, D., Martinez, K., Gómez-Pérez, A.: A Semantically Enabled Service Architecture for Mashups over Streaming and Stored Data. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 300–314. Springer, Heidelberg (2011)
29. Mitchell, T.M., Betteridge, J., Carlson, A., Hruschka, E., Wang, R.: Populating the Semantic Web by Macro-reading Internet Text. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 998–1002. Springer, Heidelberg (2009)

30. Kawamura, T., Mishiro, N., Ohsuga, A.: Green-Thumb Phone: Development of AR-based Plant Recommendation Service on Smart Phone. In: Proc. of International Conference on Advanced Computing and Applications, ACOMP (2011)
31. Kawamura, T., Shin, I., Nakagawa, H., Nakayama, K., Tahara, Y., Ohsuga, A.: ONTOMO: Web Service for Ontology Building - Evaluation of Ontology Recommendation using Named Entity Extraction. In: Proc. of IADIS International Conference WWW/INTERNET 2010, ICWI (2010)
32. Kawamura, T., Nagano, S., Inaba, M., Mizoguchi, Y.: Mobile Service for Reputation Extraction from Weblogs - Public Experiment and Evaluation. In: Proc. of Twenty-Second Conference on Artificial Intelligence, AAAI (2007)