Implementing Customer Reception Service in Robot Cafe using Stream Reasoning and ROS based on PRINTEPS

Takeshi Morita, Yu Sugawara, Ryota Nishimura, and Takahira Yamaguchi

Faculty of Science and Technology, Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
{t_morita, nishimura, yamaguti}@ae.keio.ac.jp
http://www.yamaguti.comp.ae.keio.ac.jp/

Abstract. We have developed PRactical INTElligent aPplicationS (PRINTEPS) which is a platform for developing comprehensive intelligence applications. This paper introduces an application of PRINTEPS for customer reception service in robot cafe by using stream reasoning and Robot Operating System (ROS) based on PRINTEPS, and for integrating image sensing with knowledge processing. Based on this platform, we demonstrate that the behaviors of a robot in a robot cafe can be modified by changing the applicable rule sets.

Keywords: ROS, Stream Reasoning, PRINTEPS, SWRL

1 Introduction

Designing machine-human task collaboration often requires integration of the image sensing technologies that help recognize surrounding circumstances by using a rule set of a target operation. However, a major hurdle exists in connecting the two directly. This is because a huge grain-size difference exists between information acquired through image sensing and that expressed by a rule set.

As a means to achieve integration between rules and image sensing, we propose a novel method to integrate signals (dynamic sensing data) and symbols (RDF stream) via Robot Operating System (ROS) [2] and stream reasoning tool C-SPARQL [1] based on PRINTEPS [1] [4]. In existing approaches (e.g. KnowRob [3]), the integration of image sensing and knowledge processing is achieved simply by adding knowledge expressions such as conceptual information to the object models. On the other hand, this study attempts to integrate dynamic information (e.g., people involved in time-series changes) acquired through image sensing with static information (ontologies and business rules) by using C-SPARQL.

We conducted a case study of a robot cafe customer-reception service using Pepper [2], which is an emotion-recognizing humanoid robot. As a result, a robot

[1] http://printeps.org/index_en.html
[2] https://www.aldebaran.com/en/cool-robots/pepper

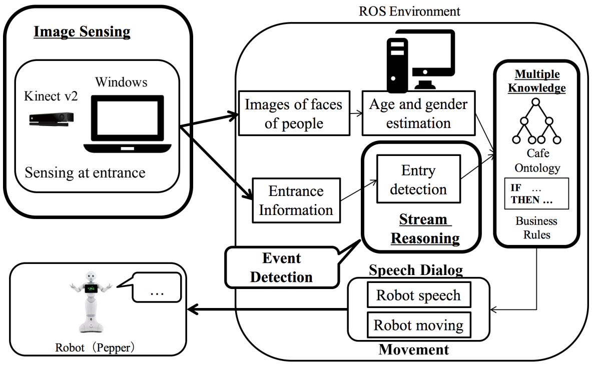

Fig. 1. System outline of customer reception service in robot cafe.

can use the rule sets that the person has, and we can construct a more efficient robot service system. A demo movie for the customer reception service can be seen on YouTube [3].

2 System Outline

Figure 1 shows the system outline of customer reception service in robot cafe. A Windows machine with a kinect v2 collects the results of sensing at the cafe entrance to the ROS environment via UDP or socket communication. Then, the information sent is analyzed based on stream reasoning, an event of entry is detected, and the age and gender of the persons are judged. The information on the event or its attribute acquired in this process is used for knowledge processing, and a robot motion program runs based on the result acquired during knowledge processing. The robot moves and speaks out according to the instructions from the business rules. With such a series of processes, the robot responds to visitors at their arrival.

3 Customer Reception Service in Robot Cafe

The customer reception service mainly consists of the “customer detection” process which detects an incoming customer by means of the Kinect sensor and the “greeting to the customer” process which orders the robot Pepper to give the customer a greeting based on the business rules described by semantic web rule langugage (SWRL).

3.1 Customer Detection Process

Figure 2 shows the C-SPARQL query for detecting a customer. This query measures at every one-second interval the distance between the face of every person

[3] https://youtu.be/HbHHT2F2Cvo

```
REGISTER QUERY CustomerDetectionQuery AS
PREFIX f: <http://larkc.eu/csparql/sparql/jena/ext#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX printeps: <http://printeps.org/cafe/entrance/>

SELECT ?s (COUNT(?s) AS ?cnt) (AVG(?distance1) AS ?distance)
FROM STREAM <http://printeps.org/cafe/entrance> [RANGE 3s STEP 1s]
WHERE {
  ?s rdf:type printeps:Customer;
     printeps:positionAtTime ?ts1, ?ts2.
  ?ts1 printeps:distance ?distance1. ?ts2 printeps:distance ?distance2.
  BIND(?distance2 - ?distance1 AS ?difference)
  FILTER((f:timestamp(?s, printeps:positionAtTime,?ts1)
    > f:timestamp(?s, printeps:positionAtTime,?ts2)) && 0.1 < ?difference)
}
GROUP BY ?s HAVING (AVG(?distance1) < ?distance_to_entrance && 1 < COUNT(?s))
```

Fig. 2. C-SPARQL query for detecting customers.

(detected within 3 s) and Kinect. The distances between the faces of the people detected within 3 s (ID: c1) and Kinect are chronologically shown as c1d1, c1d2, c1d3, and the values of c1d1-c1d2, c1d1-c1d3, and c1d2-c1d3, are computed in order to count the number of values greater than 0.1. The value 0.1 is determined based on the accident error of Kinect's depth sensor value. This measurement is used to avoid erroneously detecting someone who is in front of the cafe but is not approaching it as a customer. A count exceeding 1 means that someone is approaching the cafe (Kinect), which is a measurement used to avoid erroneously reacting with a customer who is leaving the cafe. The average value of the distances between the faces of the people detected within 3 s and Kinect is also calculated, and if the average value is less than the ?distance_to_entrance (3.3 m), the customers are recognized as having entered the cafe, and their IDs, the aforementioned count, and the average values are returned. The value of ?distance_to_entrance is obtained using a SPARQL query for measuring a distance between Kinect and an entrance by predefining, based on the cafe ontology, the distance between the location in which Kinect is installed and the entrance of the cafe. Currently, a distance between a person's face and Kinect is the only information that is used. However, if the sensor can obtain various attribute information from a person in the future, more complex customer detection based on such information (e.g., discerning a customer from a cafe clerk based on clothing) will be realized.

3.2 Greeting to the Customer Process

Figure 3 shows the rule that the module applies when determining a greeting statement upon detecting two customers. In this study, we used the reasoning engine Pellet [4] to apply the greeting properties and their values to the robot-class instance and to determine a greeting statement. The rule shown in Figure 3

[4] http://clarkparsia.com/pellet

```
Entrance(?entrance), GreetingAtEntrance(?service),
Group(?group), Pepper(?pepper), hasGroup(?entrance, ?group),
robotPosition(?pepper, ?entrance), servedBy(?service, ?pepper),
servedTo(?service, ?group), numberOfCustomers(?group, 2)
-> greeting(?pepper, "Welcome. Two seats?")
```

Fig. 3. Example of greeting statement rule.

means that a group of two people is found at the entrance; Pepper is positioned at the entrance; when Pepper provides a service to the group, Pepper gives the statement: "Welcome. Two seats?"

If a cafe owner wants a robot to give a more detailed greeting statement based on a customer's age, gender, members of the party (e.g., family members, girl or boyfriend), the owner must only revise the rule defined there.

4 Future Works

In the future, we will try to make more types of information available by using stream reasoning for more image sensing processes to make the services more effective, and to enable easier correction of the system, such as by clarifying how many processes are needed for actually changing the service.

Acknowledgement

We are grateful to Mr. Yusuke Nakayama of the Graduate School of Science and Technology of Keio University for his software-implementation for this study.

This study has been supported by "A Framework PRINTEPS to Develop Practical Artificial Intelligence" of CREST of JST.

References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: A Continuous Query Language for RDF Data Streams. International Journal of Semantic Computing 4(1), 3–25 (2010)
2. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
3. Tenorth, M., Beetz, M.: Knowrob: A knowledge processing infrastructure for cognition-enabled robots. International Journal of Robotics Research 32(5), 566–590 (2013)
4. Yamaguchi, T.: A platform printeps to develop practical intelligent applications. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers. pp. 919–920. UbiComp/ISWC'15 Adjunct, ACM (2015)