2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (UR)
August 19~22, 2016 at Sofitel Xian on Renmin Square, Xian, China

A System Architecture to Control Robot through the Acquisition of Sensory Data in IoT Environments

Yoosang Park, Jongsun Choi, and Jaeyoung Choi
School of Computer Science and Engineering, Soongsil University, 369 Sangdo-Ro, Dongjak-Gu, Seoul 06978, KOREA
(Tel : +82-2-824-3862; E-mail: {yoosang.park, jongsun.choi, choi}@ssu.ac.kr)

Abstract - Sensory data should be converted into meaningful situational information so that context-aware system can provide various robot services for users on IoT environments. Context-aware system gathers sensor data from the natural environment, which is converted to sensory data and abstracted as situational data to provide robot services. On robot runtime environment, context-aware system is required to access information of robot devices, description of robot functions, and list of robot services, in order to control robot tasks. However, those two different systems should be loosely coupled. In order to adapt the abstract concept on robot runtime environments, the sensory data acquisition system is required. In this paper, we propose system architecture to control a robot based on the acquisition of sensory data. This architecture includes the sensor data conversion process to provide appropriate control information for robot tasks.

Keywords – IoT Environments, Robot Services, Acquisition of Sensory Data, Situational Information

1. Introduction

With the advent of robot hardware such as humanoid robots and intelligent service robots, many studies related to control these robots have been actively carried out [1, 2]. In order to control robots, we need robot control description and system resource management to recognize the robot tasks. All hardware and software resources on robot platform are referred as system resources.

In order to exploit system resources effectively, following two aspects should be carefully considered. The first one is capability of the system resource [3]. Due to the volume and cost constraints, the system managed resources in robot hardware platform are limited. Second is the utilization cost for IoT sensors interacting with robots. Considering the cost of interaction with other resources, connection cost for each IoT sensor is the most expensive.

Many researches on software platform have been conducted to figure out the capability of system resources. In this context, software platform refers to DARwIn-OP [5], Robot Operating System (hereinafter 'ROS') [6], and Microsoft Robotics Developer Studio [7]. These software platforms support robot task abstraction, communications between robot tasks, and utilization policies for system resources. Also, they provide custom plug-ins to offer various management facilities.

However, there might be problems such as utilization of heterogeneous devices and expensive connection cost. In order to process sensory data in robot environments, both of these problems should be considered. System architecture with sensory data acquisition is required to provide robot tasks. Sensors in IoT environments have the following features: heterogeneous, different types of sensory data, and different types of intermediate value formats. Repeated connections of sensors with components of software platforms delays process routines and robot tasks; therefore, sensory data acquisition should logically be separated from robot software platform.

In this paper, we propose a system architecture which allows robot software platform to adapt other resources like IoT sensors in ROS. The system architecture includes a structure for separating sensory data acquisition from robot software platforms to decreases the connection cost for accessing IoT sensors. Therefore, robots are able to get control information without any attempt to connect IoT sensors directly.

2. Preliminary

ROS provides publish/subscribe communication topology to control robot tasks [8]. All tasks in ROS are invoked by topics with messages. Topics are delivered to every node by ROS master node through best-efforts, therefore each task connections are logically isolated. As ROS is an open-source project, packages consisting of several sets of tasks and codes, are offered to developers.

In order to handle robot services based on sensory data, the data should be abstracted and converted to information which the robot can understand [9]. Many studies have been conducted to process the sensory data to solve problems like heterogeneous devices and intermediate value formats. Procedure to convert sensory data into environmental information includes sensor access, acquisition of values, and conversion of situational information.

Rosbridge [10], is an interface node to connect ROS system. Rosbridge runs on ROS as a node. External programs can connect and communicate. This concept gives an advantage that it does not require to write codes for the nodes in a form of ROS. Furthermore, topics defined in system requirements can be used through Rosbridge.

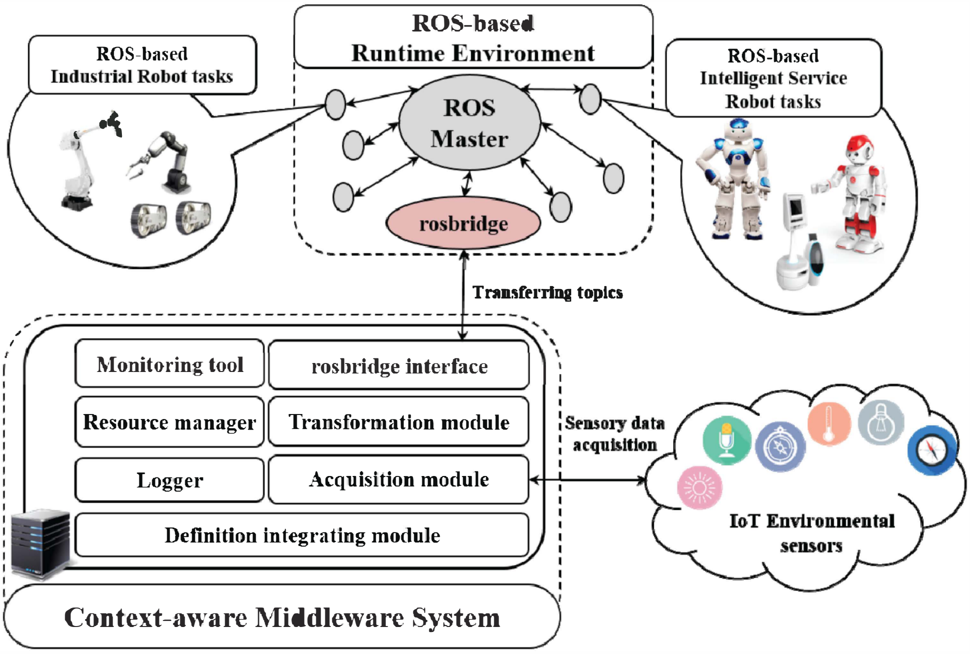

Fig. 1. A conceptual system architecture for providing robot tasks with sensory data acquisition

3. Related Works

In this section, we examine the former works related to the application and management of sensors in IoT environments.

One of the researches for providing context-aware Robot as a Service (RaaS) [11] suggests a system architecture using approaches such as redundant execution, redundant voting, centralized cloud server, and single service broker.

COMOR [12], an industrial robot system, provides ontology-based descriptions, and it operates robot tasks with multi-agents, in which each agent collects environmental information. This system includes a component which uses Bayesian reasoning to eliminate redundant environmental information.

Sensor fusion based human detection and tracking system [13] provides definitions of robot tasks as finite state machine and it uses environmental information as state transition. Finite state machine-based system architecture guarantees every unit to be exclusive, therefore there is the least consideration about robot task redundancy.

4. Suggested System Architecture

4.1 Conceptual System Architecture

Figure 1 shows conceptual system architecture to control robot tasks through sensory data acquisition on IoT environments. The suggested system consists of robots running on ROS environment, sensory data acquisition module, and heterogeneous sensors in IoT environment.

ROS-based robots are initiated by receiving topics from ROS master node. Every topic identifies its destination node. These topics contain messages to be used by the nodes. The main idea of this paper is that environmental information needs to be abstracted by sensory data acquisition system, then it is converted to topic-based information, finally Rosbridge is used to avoid violations of ROS runtime environment and communication protocol. Figure 2 shows a flow from sensor data to robot tasks.

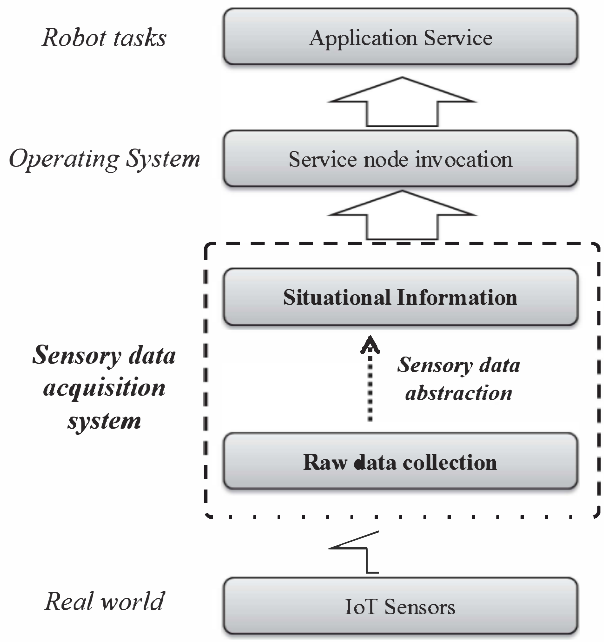

Fig. 2 Procedure for providing robot tasks with sensory data acquisition

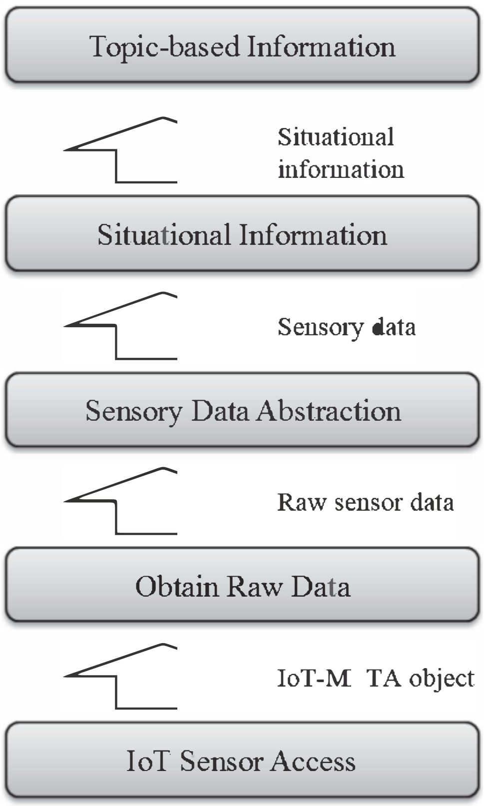

Fig. 3. Phases for sensory data abstraction

In order to receive topic-based information from IoT sensory data or control information, the ROS master node requires additional messages for topics; this is performed by Definition Integrating Module as shown in Fig. 1. As the module requires available topic lists, it asks Rosbridge to deliver the lists. These lists provide descriptions which can be used by developers to configure IoT sensors. The suggested system returns an adjusted list to the ROS master node for application.

4.2 Processing IoT Sensory data

Figure 3 shows five phases for abstracting sensory data. They are IoT sensor access, obtain raw data, sensory data abstraction, situational information, and topic-based information. The details of each phase are as follows:

A. IoT Sensor Access Phase

In order to access heterogeneous sensors, access information is required. Therefore, a specific integration scheme for heterogeneous device profile should be designed. This scheme is referred as IoT-META object. It contains connection method, access address, types of raw sensor data, and equation for converting sensor values to sensory data. Through this phase, each profile data is converted to an IoT-META object.

B. Obtain Raw data Phase

The main objective of this phase is obtaining values from IoT sensors using IoT-META objects from phase A. This phase also eliminates the duplicated access information to avoid redundancy issue. This phase obtains raw sensor data from each IoT-META object.

C. Sensory data Abstraction Phase

Raw sensor data is converted to intermediate information data, and then it is transformed into situational information. In Phase C, the raw sensor data is converted to sensory data that leads to intermediate information data using equation from IoT-META object.

D. Situational Information Phase

Surroundings, referring to situational information, indicate certain environment status that cannot be represented by a single sensor. Phases from A to C, however, are about obtaining sensory data from many heterogeneous sensors and converting them to intermediate information data. In order to transform each data from intermediate information data into situational information, a composing procedure should be executed. A mapping table for situational information is required for the procedure. It contains information to categorize the intermediate data and derive an appropriate situational information.

E. Topic-based Information Phase

In order to control robot tasks based on surroundings, phases from A to D are processed with IoT sensors. Since ROS nodes are invoked by topic-based information, situational information should be transformed to topic-based information. To figure them out, a syntax analyzer is also required that translates situational information into topic-based information. Finally, suggested system sends the topic-based information to Rosbridge.

Through these phases for sensory data abstraction, each sensory data can be converted into topic-based information to provide robot tasks in ROS.

5. Conclusion

In this paper, we proposed system architecture to control robot tasks through sensory data acquisition on ROS environments. The system aims to solve the problems related to sensory data acquisition and heterogeneous IoT devices. This suggested system performs sensory data acquisition through logical isolation. As a result, robot tasks are controlled by topic-based information generated by sensor data abstraction. Therefore, IoT devices become easily adaptable components for ROS-based robots.

Acknowledgement

This work was supported by the Industrial Convergence core Technology Development Program (No. 10048474) funded by the Ministry of Trade, Industry & Energy (MOTIE), Korea.

References

[1] R. A. El-laithy, J. Huang and M. Yeh, "Study on the use of Microsoft Kinect for robotics applications," Position Location and Navigation Symposium, pp. 1280-1288, 2012.

[2] L. A. Grieco, A. Rizzo, S. Colucci, S. Sicari, G. Piro, D. Di Paola, and G. Boggia, "IoT-aided robotics applications: Technological implications, target domains and open issues," Computer Communications, Vol. 54, pp. 32-47, 2014.

[3] R. Seiger, C. Seidl, U. Aßmann, and T. Schlegel, "A Capability-based Framework for Programming Small Domestic Service Robots," Proceedings of the 2015 Joint MORSE/VAO Workshop, ACM, pp. 49-54, 2015.

[4] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," Autonomous Robots, Vol. 22, Issue 2, pp. 101-132, 2007.

[5] I. Ha, Y. Tamura, H. Asama, J. Han and D. W. Hong, "Development of open humanoid platform DARwIn-OP," 2011 Proceedings of the SICE Annual Conference, IEEE, pp. 2178-2181, 2011.

[6] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," ICRA workshop on open source software, Vol. 3, No. 3, 2009.

[7] J. Jackson, "Microsoft robotics studio: A technical introduction," Robotics & Automation Magazine, IEEE, Vol. 14, Issue 4, pp. 82-87, 2007.

[8] S. Cousins, "Welcome to ROS Topics [ROS Topics]," Robotics & Automation Magazine, IEEE, Vol. 17, Issue 1, pp. 13-14, 2010.

[9] Seng W. Loke, "Context-Aware Artifacts: Two Development Approaches," Pervasive Computing, IEEE, Vol. 5, No. 2, pp. 48-53, 2006.

[10] C. Crick, G. Jay, S Osentosiki, B. Pitzer and O. C. Jenkins, "Rosbridge: ROS for Non-ROS Users," Proceeings of the 15th International Symposium on Robotics Research, pp. 339-346, 2011.

[11] Y. Chen and H. Hu, "Internet of intelligent things and robot as a service," Simulation Modelling Practice and Theory, Vol. 34, pp. 159-171, 2013.

[12] T. Stipancic, B. Jerbic and P. Curkovic, "Robotics and Computer-Integrated Manufacturing," Vol. 37, pp. 79-89, 2016.

[13] C. Wang, S. Tseng, P. Wu, Y. Xu, C. Liao, Y. Lin, Y. Chiang, C. Lim, T. Chu and L. Fu, "Human-Oriented Recognition for Intelligent Interactive Office Robot," 13th International Conference on Control Automation and Systems, IEEE, pp. 960-965, 2013.