ROS-based Integration of Smart Space and a Mobile Robot as the Internet of Robotic Things

David Uchechukwu[1], Arslan Siddique[2], Aizhan Maksatbek[3] and Ilya Afanasyev[2]
[1]Kazan Power Engineering University, Kazan, Russia
[2]Innopolis University, Innopolis, Russia
[3]Yildiz Technical University, Istanbul, Turkey
uched41@yahoo.com, a.siddique@innopolis.university, aizhanmaksatbek@gmail.com, i.afanasyev@innopolis.ru

***Abstract*—The Internet of Robotic Things is a concept that is rapidly gaining traction in the robotics industry. As the field of robotics advances, one of the obstacles to its widespread adoption remains the high cost of purchasing and maintaining robot. Although the gap is continuously closing, robots these days cannot make decisions with the efficiency that humans can. People are spatially aware, and we can perceive and understand changes in the environment that robots are not capable of. This paper attempts to propose how a smart space can be implemented to increase the spatial awareness of a robot by providing more data to make better informed decisions. We focus on using the Robot Operating System (ROS) as a framework to integrate Smart Space and a mobile robot to expand the robots sensory information and make collision avoidance decisions.**

I. Introduction

The Internet of Robotic Things (IoRT) is a relatively new field, the concept of which has attracted considerable attention from the research community and industry. It carefully combines the individual aspects of cloud robotics and the Internet of things, capable of forming a new field with an estimated potential market of $21 billion by 2022 [1]. ABI Research introduced this novel field in the detailed report [2]. They showed how key features of robotics technology, namely movement, manipulation, intelligence and autonomy, can be enhanced by IoT scenarios. The simple illustration of the IoRT architecture is shown in Fig. 1, where the intelligent agents: Robots and Smart Space interact with each other through the Network Communication Protocol. In the study [3], Partha Ray presented the architectural components and research challenges in this area, conclusively describing IoRT as:

> ”A global infrastructure for the information society enabling advanced robotic services by interconnecting robotic things based on, existing and evolving, interoperable information and communication technologies where cloud computing, cloud storage, and other existing Internet technologies are centered around the benefits of the converged cloud infrastructure and shared services that allows robots to take benefit from the powerful computational, storage, and communications resources of modern data centers attached with the clouds, while removing overheads for maintenance and updates, and enhancing independence on the custom cloud based middle-ware platforms, entailing additional power requirements which may reduce the operating duration and constrain robot mobility by covering cloud data transfer rates to offload tasks without hard real time requirements.” [3]

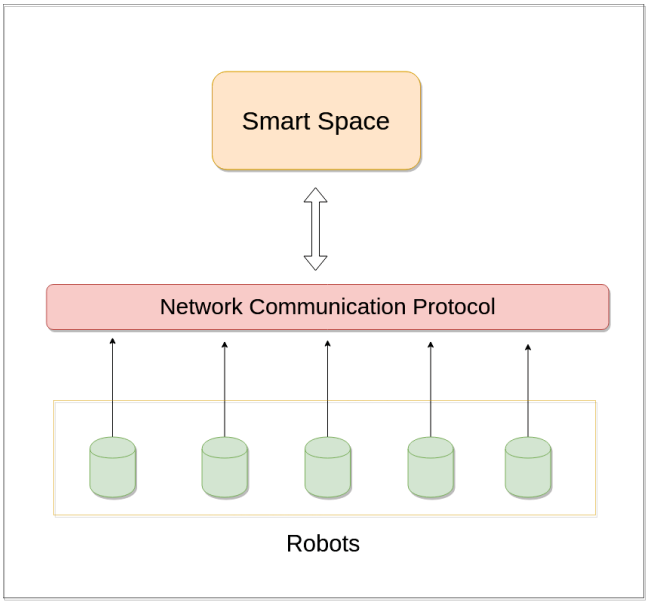

Fig. 1. The simple illustration of the Internet of Robotic Things (IoRT), where the intelligent agents: Robots and Smart Space interact with each other through the Network Communication Protocol architecture

The smart space utilizes a distributed information processing model to provide support to requesting devices. This additional data source will enable the robot to make better informed decisions. It also eliminates the need to over-equip smaller robots with non-critical sensors and equipment. In our experiment, we leveraged the smart space methodology to provide computer vision support to low-end robot with no computer vision hardware.

The key contributions of this research paper are as follows:

- The demonstration of how Smart Space with robust off-board hardware can assist simple robots with no computer vision and image processing capabilities to take complex and computationally expensive decisions.
- The development of the method to replace large number of on-board noisy robot sensors with fewer external sensors.

- The implementation of the ground framework for the field of Internet of Robotic Things (IoRT).

The rest of the paper has been organized as follows: Section II describes the research work related to our project. Section III explains our methodology and algorithm. Section IV contains of the experimental setup where we give the details on our mobile robot node and Smart Space node. Section V presents the experimental results and discussion, and finally we conclude in Section VI.

II. Related Work

Smart Spaces are actively implemented in areas such as connected homes, smart cities, sensor networks, smart factories, services of commercial and office real-estate, etc. [4], [5]. What is more, Smart spaces often support services that are configured using "smart applications" that actively include surrounding digital devices, sensors, controllers, routers and cloud services [6]. At the same time, the number and heterogeneity of IoT devices create challenges in their management, therefore researchers have been focusing on the middleware architecture and orchestration, which could become the standard for the software producers [7]. Using Smart Space technology, companies are able monitor environmental sensor data in real time mode and improve the awareness of intelligent agents (such as robots or autonomous systems) [8]. For this purpose, researchers have been developing conceptual models of a cyberphysical environment based on special approaches to the distribution of sensory, network, computing, control, information and service tasks between mobile robots, embedded devices, stationary service equipment, clouds and information resources [9].

Smart space can also be used in multi-agent robotic systems and swarm robotics, where a group of robots work together to accomplish a common task or group of tasks. In such tasks, it can be costly to equip each individual robot with a full set of sensors that it needs. Therefore, Smart Space can guarantee that all robots will receive the necessary data from its sensors. In this way, IoRT can demonstrate the integration of robotic systems and IoT, providing new technological solutions using wireless sensor networks (WSN), cloud computing, distributed planning, management and even ledgers [8]. In Swarm Intelligence studies, the behavior of complex dynamic systems is modeled as a swarm of digital telecommunications networks (nerves), ubiquitous embedded intelligence (brain), sensors and tags (sensory organs), and software (knowledge and cognitive competence) [10].

Although in this work we focus on the use of Robot Operating System (ROS) as a framework for connecting Smart Space and a robot for a centralized decision-making on avoiding collisions, in present there are investigations that offer protocols of asynchronous communication between robots and end-users via the cloud (such as ROSLink [11] and MAVLink [12]).

The integration of smart space with mobile robots as moving agents can provide a number of advantages for logistics, service, domestic and assistive robotics applications [8], [13]. Moreover, the Internet of Robotic Things (IoRT) is closely linked with the Industry 4.0, where Industrial Internet aims to computerize the manufacturing industry for giving manufacturing robots the ability to think on their own and make decisions with little or no human input [14]. Similar proposals for the development of a Smart Environment has also been discussed in [8] that describes the components of IoRT architecture. IoRT architecture consists of three main layers as demonstrated also in Fig. 2. The first layer is the physical layer which includes robots as intelligent agents and sensors for monitoring the agents and environment parameters. The network and control layer includes communication protocols, Internet protocols (IP), routers and cloud data servers. It establishes communication between the other two layers. At the service and application layer, user-friendly programs are implemented to monitor and control physical layer components via communication protocols.

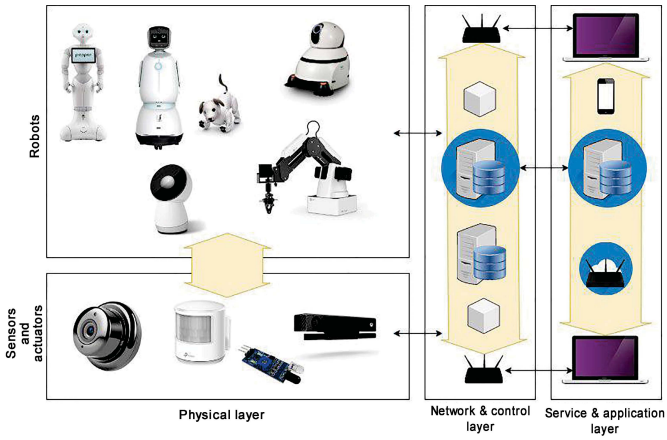

Fig. 2. Internet of Robotic Things (IoRT) architecture. (Some figure elements are courtesy of Hayward Industries, LG Electronics Inc., Service Robots under Warrington Robotics Ltd, KUKA AG)

The differential drive mobile robot, Lego EV3, which we used in our experiments, has been described and used extensively in robotics research [15], [16]. This mobile robot communicates to a ROS node running on a host computer. A camera is also attached to the host computer for monitoring and surveillance of the mobile robot. The robot location can be identified using April tags [17], [18] placed on the robot platform. Authors [19] have described the characteristics and key features of widely used markers such as ARTag, AprilTag and CALTag. AprilTag markers seem to be well developed, fast and easy to use that is why they have been utilized in this project. As the robot detects an object in front of it using its ultrasonic sensor, it sends request to the ROS node to identify surrounding objects.

The field of object detection and classification is a very well studied problem in Computer Vision because of its use in several areas such as smart surveillance systems and tracking. However, this task remained a difficult problem for a long time

due to the variation in human visual appearance, illumination variance, intermixing of humans, variation in human physique, loss of 3D information because of taking a 2D picture. At earlier times, researchers used to handcraft the characteristic of object features and used some classifier to identify the object [20]. With the rise of deep learning, Convolutional Neural Network [21] became a popular approach for this task. Large datasets were collected and made available publicly available. Region based Convolution Neural network (R-CNN) methods which includes R-CNN [22], Fast R-CNN [23] and Faster-RCNN [24] developed by Microsoft research team are very popular. Another family includes You Only Look Once (YOLO) [25], YOLOv2 [26] and YOLOv3 [27]. Among all these techniques, YOLOv3 is the most robust and fastest approach which is why this technique has been chosen in this project.

III. Methodology

A. Components

We will briefly describe some of the key components of our experimental setup.

- ROS (Robot Operating System)
- AprilTag for Object Location.
- Obstacle Detection and Classification

1) ROS (Robot Operating System: ROS is a robotics middleware that provides libraries and tools for creating robot applications [28]. It is widely used in both commercial and open source robotics projects thanks to its free BSD license. ROS enables robotics application to be build relatively quickly by providing standardized functionalities for controlling and communicating with the robot hardware. One of the core principles of ROS is the peer to peer architecture [29]. This enables each component to be built and act independently. These individual components can then communicate with each other using a network infrastructure centered around the ROS Master. This peer to peer architecture is a perfectly suited for the smart space implementation.

2) AprilTag for Object Location: AprilTag is one of the most widely used visual fiducial markers. Fiducial markers simply serve as a reference to the location of an object in an imaging system. AprilTag is used to detect the location and orientation of an object in a given field of view. AprilTag draws some similarities from the QR-Code and 2 dimensional bar codes. It achieves long range detection by encoding smaller data payloads [18]. The aprilTag has been shown to offer better accuracy and detection rates when compared to other fiducial markers like ARtags [17].

3) Obstacle Detection and Classification: We implemented the widely used YOLO classification algorithm. Satisfactory results were acheived using YOLOv3 with classification time of less than 2 seconds. The implemented YOLOv3 method can classify 80 distinct objects accurately within 2 seconds. We utilized YOLOv3 weights and architecture trained on COCO dataset into OpenCV's Deep Neural Network Module (DNN) for object recognition task. However, OpenCV version must be greater than 3.4.2 for this method because older versions cannot load YOLO into their DNN module. The implemented object detection and classification method has been described in pseudo-code of the Algorithm 1.

Algorithm 1: OpenCV based YOLOv3 object detection method

```
Input: 3d Numpy image array
Output: List of objects recognized in the image
Initialize thresholds for weak detections
Load output label names and pre-trained YOLOv3
    architecture and weights into readNetFromDarknet
    method of OpenCV DNN module
Forward pass blobs from the image into YOLO object
    detector.
for each layer output do
    for each bounding box detected do
        extract class ID and confidence
        if confidence > threshold then
            extract bounding box coordinates and IDs
Apply non-maxima suppression to filter out remaining
    weak and overlapping detections
Extract output label class for each final detection and
    return the list
```

B. Architecture

Well defined architectural models for IoRT can be seen in [3]. In our work, we leveraged the smart space methodology to provide computer vision capabilities to a Lego EV3 robot. Our smart space is implemented as a ROS node.

The aim of our experiment is to prove that simple robot without computer vision and image processing hardware can take advantage of the information provided by the smart space to make data-driven decisions. A robot equipped with an only ultrasonic sensor can detect obstacles but is not able to figure out what the obstacle is. An ultrasonic sensor works by emitting high frequency sound waves and measuring the time it takes for the signal to return [30]. By doing some basic calculations using the speed of sound, we are able to determine the distance to the nearest obstacle. An ultrasonic sensor is not able to provide an information on the characteristics of the detected object. Normally, when a robot detects an obstacle, the next action is to avoid this obstacle by changing direction. In reality this obstacle may not pose a serious enough threat to warrant a change in direction. Using the example of a cleaning robot which uses its ultrasonic sensor to detect an obstacle. A visualization of the data returned by the ultrasonic sensor is shown in Fig. 4 [31]. The red area represents obstacles closer than a specified setpoint. This obstacle could be an empty plastic bottle which poses no serious threat to the robot in

which case the robot should be able to continue on its path simply pushing the water bottle aside. A more rigid object like a vertical pole will return the same or similar ultrasonic data to the robot. But in this case, the robot should change its path because it is not able to move the object.

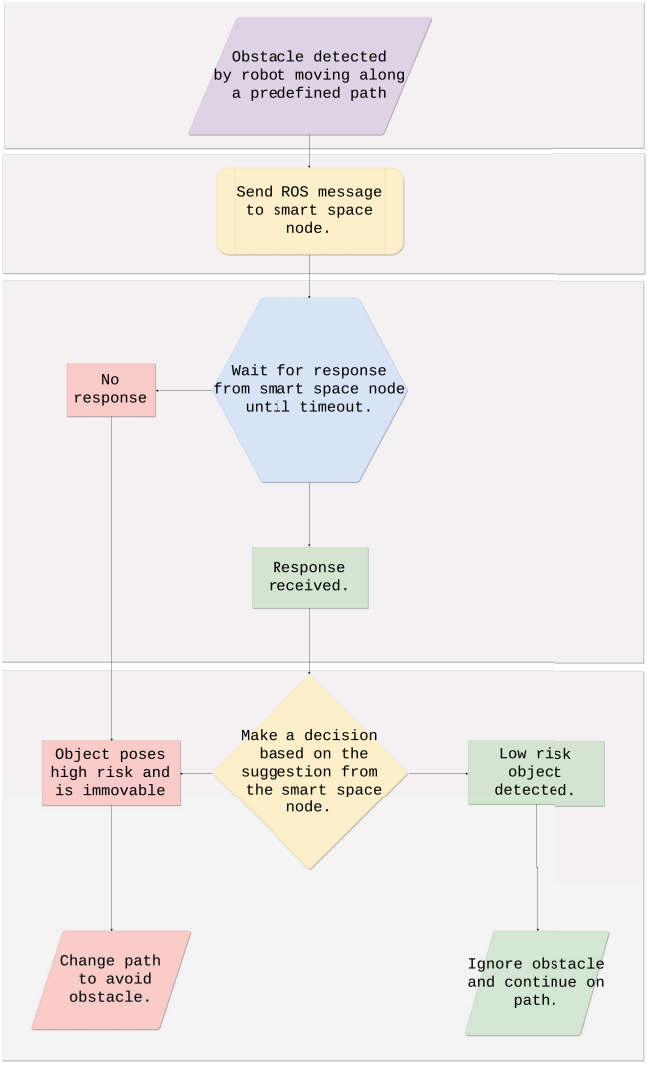

Fig. 3. The block-scheme of proposed methodology to implement smart spaces for Internet of Robotic Things (IoRT) architecture

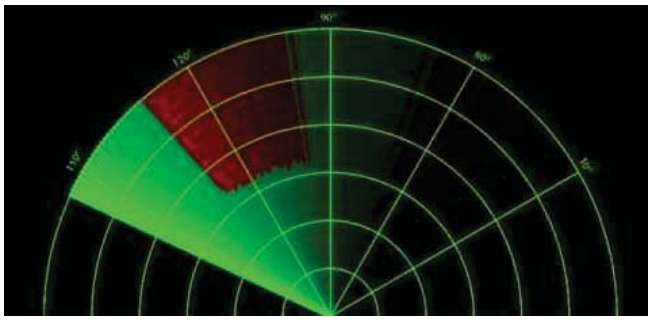

Fig. 4. Image visualization of the data obtained by an ultrasonic sensor [31]

In our experiment, We implemented the smart space as a ROS master which is equipped with an IP camera to provide live video stream of the environment while the robot runs a ROS node which sends and receive messages from the ROS master. An AprilTag is attached to the robot for determining the location and tracking of the robot. The robot requests information from the smart space node by sending a ROS message. This message contains information needed to locate the robot. On receiving this message, the smart space locates the robot using the provided information and attempts to classify obstacles in the vicinity of the robot. This information in then returned to the robot, enabling it to make data-driven decisions. The Fig. 3 shows athe block diagram that explains our methodology.

IV. Experimental setup

As shown in the Fig. 5, the main components of our experimental setup are:

- Mobile Robot's Node.
- Smart Space's Node.

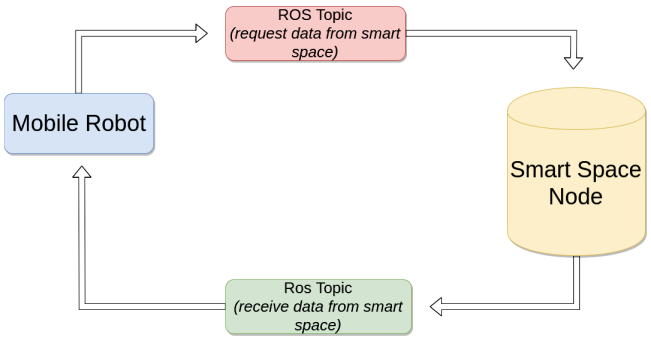

Fig. 5. The main components of our experimental setup: Mobile Robot's Node, Smart Space's Node and ROS Topics

Communication between the robot and the processing node achieved using ROS. One great feature of ROS is that nodes (independent components of the robot) can communicate with each other using messages. We leveraged this framework to send messages from the mobile robot to our smart space processing device.

A. Mobile Robot Node

TABLE I. Specification of mobile robot

System	Configuration
Hardware	Lego Mindstorm EV3
Processor	TI Sitara AM1808@300 MHz
RAM	64 MB
Hardware memory	16 MB

The hardware and software specifications of our mobile robot have been presented in the Table I. The lego EV3 does not officially support ROS. ROS can also not be installed on a running ev3 image because the lego ev3 does not have the

hardware resources required to compile and install ROS. Using the official ev3 image, we built a custom image to include ROS. This image was built using Docker. Docker is a set of tools for building and running virtual operating systems.

The AprilTag is attached on the robot as shown in Fig. 6. The robot is also equipped with an ultrasonic sensor for obstacle detection. On detecting an obstacle, the robot stops and publishes a message to outgoing message topic on which the smart space node is already listening. This message contains identification information. The identification information is the aprilTag data (Tag family and tag ID). AprilTags are divided into different families. Any combination of tag type and tag ID is guaranteed to be unique AprilTag image. Once the identification information has been sent to the smart space node, the robot starts listening for messages from the smart space node on the incoming messages topic. The smart space node locates the robot using the provided information. After doing the necessary processing, the smart space sends back data describing the obstacles in the vicinity of the robot. In our case, the smart space node replies with a list of the names of the detected objects. Based on this information, the robot can decide to continue on the predefined path or change its path. The code for the ROS node running on the Lego EV3 robot can be found in the project's Github repository [32].

Consider the example of a house cleaning robot that detects a vertical obstruction. This obstacle can be a low-threat object that can be easily moved like an empty water bottle. It could also be a table leg. If the cleaning robot is able to decipher the type of the object, it can continue on its path if the obstacle is an empty bottle, or change paths if it detects that the object cannot be moved.

```
Message {
   AprilTag_Type : Tag36h11,
   Tag_ID : 10
  }

Outgoing Topic: "/request_obstacle_data"
Incoming Topic: "/receive_obstacle_data"
```

B. Smart Space Node

The smart space node was implemented on a desktop computer whose hardware and software specifications have been presented in the Table II. The IP camera was connected to provide video stream of the environment. The smart space comprises of a ROS master which serves as a central point for all the robots in the smart space environment. The separate ROS node (smart space node) is also implemented to receive and process information from the robots.

The smart space node listens on the specified topic for messages published by the robot nodes. This message contains identification information as described above. When a message is received, the smart space node reads the video stream and

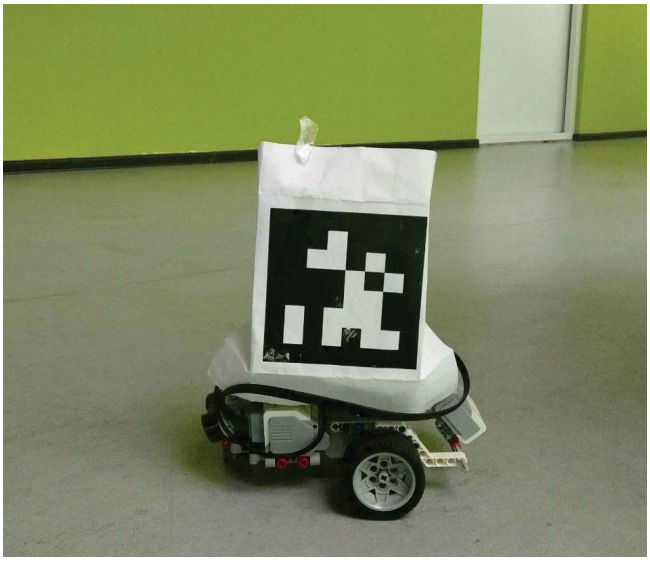

Fig. 6. The Lego Mindstorm EV3 robot with the attached AprilTag label

TABLE II. SPECIFICATIONS OF DESKTOP RUNNING SMART SPACE NODE

System	Configuration
Operating System	Ubuntu 18.04
Processor	1.7 GHz Intel Core i7
RAM	6 GB
Hardware memory	500 GB

attempts to detect all AprilTags in the field of view. We compare the tag ID and tag type of the detected AprilTags with the tag information sent by the robot in order to locate the sending robot. Using the AprilTag we can also estimate the pose of the robot which enables us to crop a specific section from the current image frame to include only the robot and the obstacle. This ensures that only the necessary part of the image is processed. The cropped section of the frame is passed to the YOLO object classifier which returns a list of the detected objects.
A name of the detected object is then packaged in a message and returned back to the robot node.

```
Message {
   objects: ["bottle"]
  }
```

ROS workspace for Smart Space node can be downloaded from the Github repository [33].

V. RESULTS AND DISCUSSION

In our experiment, we are able to detect and classify objects by leveraging the services provided by the smart space infrastructure using a low-end mobile robot with no computer vision abilities. As the mobile robot ROS node is initialized, robot starts moving forward as shown in Fig. 7.

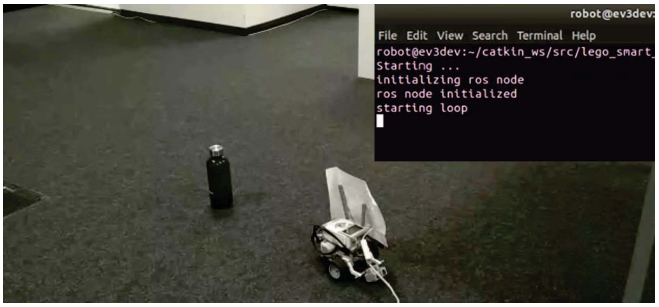

Fig. 7. When robot's ROS node is initialized, the robot starts moving forward

As, the Lego ev3 robot detects an obstacle in front of it using ultrasonic sensor, it stops and communicates to the smart space node to request for object detection as shown in Fig. 8.

Fig. 8. As the robot detects an obstacle, it stops and publishes a message to request the Smart Space's node to identify obstacle

As soon as the smart space node receives message from robot node, it reads the video stream coming from IP camera. The message published by robot contains AprilTag type and Tag ID which serve as identification information for the smart space node to find robot location. The area around robot is cropped from the current image to ensure YOLO classifier to detect only the obstacles around the robot. Using the supplied AprilTag information, the smart space node is able to locate the robot and classify the obstacle detected by the robot. Information about the obstacles is sent back to the robot node. The robot takes turn after receiving the requested information as shown in Fig. 9.

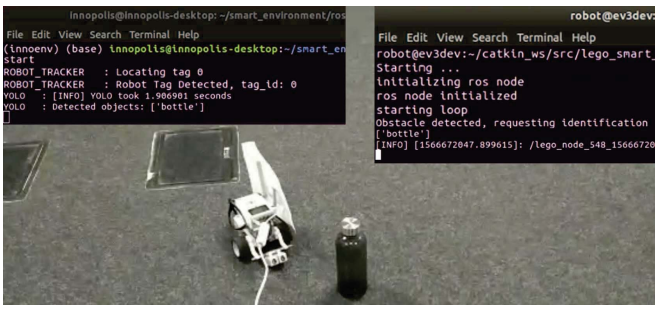

Fig. 9. Smart Space's node identifies the robot location and obstacles around the robot, and sends the list of obstacles detected back to the robot

ROS provides us several tools to understand the system and debug code. One such great tool is rqt_graph which is a GUI plugin to visualize ROS nodes and topics in the system. The rqt_graph for our experiment is shown in Fig. 10. The graph shows that there are two nodes enclosed in ellipses and two topics exchanging data between them. */lego_node* is subscribed to the topic */smart_environment/response* and */smart_environment* is subscribed to the topic */smart_environment/request*.

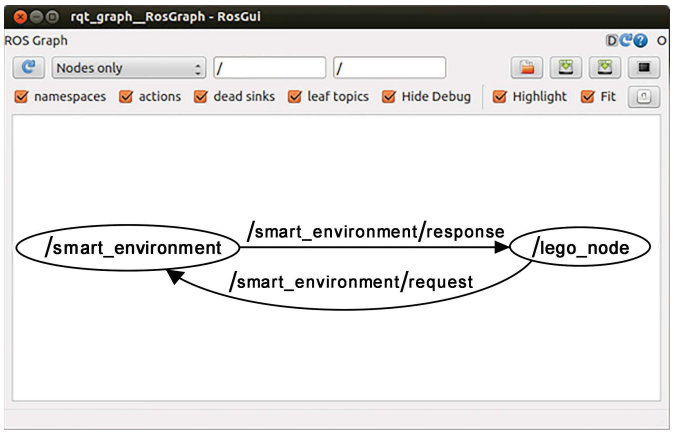

Fig. 10. The *rqt_graph* of our experiment which shows that */lego_node* is subscribed to the topic *rqt_graph/smart_environment/response* and */smart_environment* is subscribed to the topic */smart_environment/request*

When detected obstacle was classified as a low threat object (e.g plastic bottle), the robot continued on its path and pushed the obstacle away. In another test, the obstacle detected was a table. Even though a table leg and a plastic bottle will return similar data to an ultrasonic sensor, the robot was able to determine that the former is not movable and changed its path.

The whole process took less than 3 seconds, which is a decent response time for a mobile robot. Better response time can be achieved with more powerful hardware. The video demo of our experiment can be seen on YouTube [34].

VI. Conclusions

This experiment lays the ground work for a more complex smart space for the internet of robotic things. It is not always feasible to equip a robot with all the sensors and devices required for complete autonomous operation. In addition to this, data from sensors is often prone to noise or malfunction. A smart space can serve as primary or secondary source of data for robots. As a primary data source, smart spaces enable us to build minimalist robots and offload more complex but non-critical tasks to be processed on the smart space node.

More complex and fully equipped robots can take advantage of smart spaces as a secondary data source to cross-reference data which it already has, or to acquire data which the robot is currently unable to access because of its current line of sight or general situation. For example, a robot equipped with computer vision hardware and software may not be able to accurately identify an object because of its current line of sight.

A smart space can also be applied in swarm robotics, where a group of robots work together to perform a common task or group of tasks. In swarm robotics, it may not be feasible to equip each individual robot with the full range of sensors it needs. A smart space will ensure that all sensors get the required data.

REFERENCES

[1] S. Boral, "What is internet of robotic things and how does it affect you." https://www.iottechtrends.com/internet-of-robotic-things/, 2019. Accessed: 2019-09-10.

[2] D. Kara and S. Carlaw, "The internet of robotic things by abi research." https://www.abiresearch.com/market-research/product/1019712-the-internet-of-robotic-things/, 2014. Accessed: 2019-10-14.

[3] P. P. Ray, "Internet of robotic things: Concept, technologies, and challenges," *IEEE Access*, vol. 4, pp. 9489–9500, 2016.

[4] H. Liu, H. Ning, Q. Mu, Y. Zheng, J. Zeng, L. T. Yang, R. Huang, and J. Ma, "A review of the smart world," *Future generation computer systems*, 2017.

[5] M. Mazzara, I. Afanasyev, S. R. Sarangi, S. Distefano, V. Kumar, and M. Ahmad, "A reference architecture for smart and software-defined buildings," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 167–172, IEEE, 2019.

[6] D. G. Korzun, A. M. Kashevnik, S. I. Balandin, and A. V. Smirnov, "The smart-m3 platform: Experience of smart space application development for internet of things," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp. 56–67, Springer, 2015.

[7] M.-O. Pahl, G. Carle, and G. Klinker, "Distributed smart space orchestration," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 979–984, IEEE, 2016.

[8] I. Afanasyev, M. Mazzara, S. Chakraborty, N. Zhuchkov, A. Maksatbek, M. Kassab, and S. Distefano, "Towards the internet of robotic things: Analysis, architecture, components and challenges," *arXiv preprint arXiv:1907.03817*, 2019.

[9] A. Ronzhin, A. Saveliev, O. Basov, and S. Solyonyj, "Conceptual model of cyberphysical environment based on collaborative work of distributed means and mobile robots," in *International Conference on Interactive Collaborative Robotics*, pp. 32–39, Springer, 2016.

[10] O. Zedadra, A. Guerrieri, N. Jouandeau, G. Spezzano, H. Seridi, and G. Fortino, "Swarm intelligence and iot-based smart cities: A review," in *The Internet of Things for Smart Urban Ecosystems*, pp. 177–200, Springer, 2019.

[11] A. Koubaa, M. Alajlan, and B. Qureshi, "Roslink: Bridging ros with the internet-of-things for cloud robotics," in *Robot Operating System (ROS)*, pp. 265–283, Springer, 2017.

[12] L. Meier, "Mavlink micro air vehicle communication protocol: Mavlink developer guide." https://mavlink.io/en/. Accessed: 2019-10-14.

[13] P. Simoens, M. Dragone, and A. Saffiotti, "The internet of robotic things: A review of concept, added value and applications," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, p. 1729881418759424, 2018.

[14] iSCOOP, "The internet of robotic things (iort): definition, market and examples." https://www.i-scoop.eu/internet-of-things-guide/internet-robotic-things-iort/. Accessed: 2019-09-10.

[15] F. Klassner, "A case study of lego mindstorms' suitability for artificial intelligence and robotics courses at the college level," in *Acm sigcse bulletin*, vol. 34, pp. 8–12, ACM, 2002.

[16] F. Klassner and S. D. Anderson, "Lego mindstorms: Not just for k-12 anymore," *IEEE Robotics & Automation Magazine*, vol. 10, no. 2, pp. 12–18, 2003.

[17] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, IEEE, 2011.

[18] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198, IEEE, 2016.

[19] A. Sagitov, K. Shabalina, L. Sabirova, H. Li, and E. Magid, "Artag, apriltag and caltag fiducial marker systems: Comparison in a presence of partial marker occlusion and rotation.," in *ICINCO (2)*, pp. 182–191, 2017.

[20] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 1005–1009, 2013.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[26] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[27] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[29] Generation Robots, "Tutorial: Ros - robot operating system." https://www.generationrobots.com/blog/en/ros-robot-operating-system-2/, 2016. Accessed: 2019-09-10.

[30] Microsonic, "Ultrasonic technology by microsonic." https://www.microsonic.de/en/support/ultrasonic-technology/principle.html. Accessed: 2019-09-10.

[31] HowToMechatronics.com, "Tutorial: Arduino radar project." https://howtomechatronics.com/projects/arduino-radar-project/. Accessed: 2019-09-10.

[32] A. Siddique, "legonode: Ros node to move lego mindstorm ev3 robot." https://github.com/hafizas101/legoNode.

[33] D. Uchechukwu, "Smartenv: Smart information support system for internet of robotic things." https://github.com/uched41/smartenv.

[34] H. Arslan, "Smart environment demo for lego ev3 robot's obstacle avoidance." https://www.youtube.com/watch?v=zKeig5aofyg.