Feasibility Study of IoRT Platform "Big Sensor Box"

Ryo Kurazume[1], Yoonseok Pyo[2], Kazuto Nakashima[3], Akihiro Kawamura[1], and Tokuo Tsuji[4]

Abstract— **This paper proposes new software and hardware platforms named ROS-TMS and Big Sensor Box, respectively, for an informationally structured environment. We started the development of a management system for an informationally structured environment named Town Management System (TMS) in the Robot Town Project in 2005. Since then we have been continuing our efforts to improve performance and to enhance TMS functions. Recently, we launched a new version of TMS named ROS-TMS, which resolves some critical problems in TMS by adopting the Robot Operating System (ROS) and utilizing the high scalability and numerous resources of ROS. In this paper, we first discuss the structure of a software platform for the informationally structured environment and describe in detail our latest system, ROS-TMS version 4.0. Next, we introduce a hardware platform for the informationally structured environment named Big Sensor Box, in which a variety of sensors are embedded and service robots are operated according to the structured information under the management of ROS-TMS. Robot service experiments including a fetch-and-give task and autonomous control of a wheelchair robot are also conducted in Big Sensor Box.**

I. INTRODUCTION

An informationally structured environment (or an intelligent space) is a key technology for realizing a service robot in a daily-life environment in the near future. Instead of developing an intelligent robot equipped with a number of sensors, the environment itself is designed to be intelligent by structuring the information of the robot and its surroundings and providing the required information to the service robot on time and on demand. To do so, various sensors are embedded in the environment, including walls, ceilings, furniture, or objects, to acquire a variety of information, such as positions of humans, objects, and robots, or the current status of environments. Acquired information is stored in a cloud database, analyzed in cyber space, and provided to robots and users. A number of studies related to informationally structured environments have been published so far. These include "Robotic Room" [1] and "Intelligent Space" [2] at Tokyo University, "Smart Room" at MIT MediaLab [3], "Intelligent Room" at AILab [4], "Aware Home" at Georgia Tech. [5], and "Wabot House" at Waseda University [6]. Many of these are still studied actively in laboratories [7] [8] [9] [10] [11].

The current authors also started to develop a town-scale informationally structured environment named "Robot Town Project" in 2005 and have been developing a software platform named Town Management System (TMS). In 2015 we launched the latest version named ROS-TMS 4.0, which adopted the Robot Operating System (ROS) [12] as the middleware of TMS.

The idea of using a variety of sensors embedded in the environment and adapting the behavior of the system to the current situation is also called "ambient intelligence" [13]. The synergy effect of ambient intelligence and robots was studied in "CompanionAble project" [14] in the FP7 project. Additionally, in Horizon 2020, some projects related to "personalising health and care (PHC)" have been planned to utilize service robots for health-care purposes.

Although our research is closely related to ambient intelligence or smart homes [15], [16], and more recently, Internet of Things (IoT) [17] or a cyber-physical system (CPS), one of the characteristics of our approach is that we focus on the development of not only several sensor systems but also a total framework of the informationally structured system. We are also developing a software platform with high flexibility and expandability named "ROS-TMS".

II. SOFTWARE PLATFORM ROS-TMS

The CPS is a fundamental social system for the next IT generation. In a CPS, actual situations, events, and behaviors in the real world are measured by distributed sensors or obtained as open/social data. Optimum solutions are derived in the cyber world based on huge computer resources. Then, the real world is managed and controlled according to the optimum solutions given in the cyber world to realize an ideal real world. A "smart grid", which optimizes generation, transmission, and storage of electricity, is a representative example of a CPS. After the National Science Foundation (NSF) identified the CPS as a key research area in 2006 [18], this concept has been attracting much attention and a number of studies have been presented so far. To realize a CPS, IoT (Internet of Things) is one of the key technologies for gathering real-world information. In addition, to realize designed plans in the real world, robot technology (RT) is also quite important. Therefore, IoT and RT are indispensable for driving the CPS, and we refer to these two key technologies as Internet of Robot and Things (IoRT), as shown in Fig. 1.

ROS-TMS [19] is a core software platform for IoRT. We released the latest version, ROS-TMS 4.0, in September 2015

[1]Ryo Kurazume and Akihiro Kawamura are with Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan {kurazume, kawamura}@ait.kyushu-u.ac.jp

[2]Yoonseok Pyo is with ROBOTIS CO., LTD., Seoul 153-787, Korea passionvirus@gmail.com

[3]Kazuto Nakashima is with Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan k_nakashima@irvs.ait.kyushu-u.ac.jp

[4]Tokuo Tsuji is with Faculty of Mechanical Engineering, Kanazawa University, Ishikawa 920-1192, Japan tokuo-tsuji@se.kanazawa-u.ac.jp

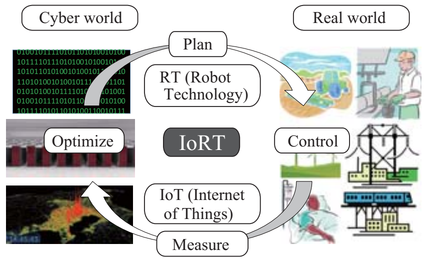

Fig. 1. Cyber Physical System (CPS) and IoRT

and proposed ROS-TMS in [20]. We briefly summarize ROS-TMS 4.0 in this section.

ROS-TMS 4.0 consists of more than 150 informationally structured nodes classified into several sensor processing modules, such as laser range finders or IC card readers, robot planning modules, task planners and schedulers, human interface modules, and database modules. The functions required for supporting life services by service robots are designed and implemented as nodes. These nodes are connected and reconfigured flexibly according to the desired service tasks. Fig. 2 shows the total structure of ROS-TMS. The summary of each module is as follows:

- TMS_DB: nodes for communicating with a database (DB Server and File Server).
- TMS_UR: nodes for communicating with users. Requests received from users are sent to TMS_TS.
- TMS_TS: nodes for planning service tasks performed by robots.
- TMS_SD: nodes for controlling sensors such as laser range finders, cameras, or RFID tag readers.
- TMS_SS: nodes for integrating sensor data.
- TMS_SA: nodes for estimating the status of the environment.
- TMS_RP: nodes for planning robot motions.
- TMS_RC: nodes for executing robot motions.

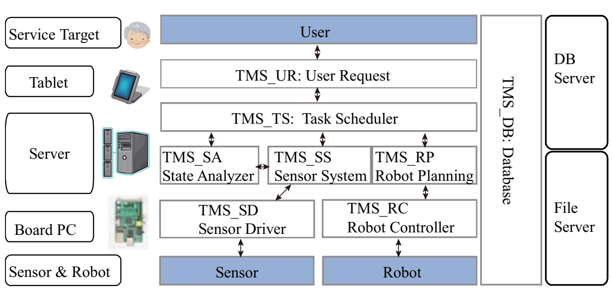

Fig. 2. ROS-TMS architecture

Among them, User Request (TMS_UR), Sensor Driver (TMS_SD), and Robot Controller (TMS_RC) modules are interface modules between users, sensors, and robots. The Task Scheduler (TMS_TS) module receives a sequence of subtasks from the Database (TMS_DB) module according to the user's requests. Task Scheduler is designed to achieve the desired task and to schedule the execution timings of subtasks. The Robot Planner (TMS_RP) module manages the execution of each subtask and sends execution commands to the Robot Controller (TMS_RC) module with proper timing.

By modularizing various sensor systems and processing algorithms based on ROS, reusability and flexibility of modules are improved and the efficiency of system development is enhanced. For example, we replaced the database from MySQL in ROS-TMS ver. 3.4.2 to MongoDB in ver. 4.0. However, most of the modules other than the TMS_DB module (interface for the database) were not influenced at all. In addition, by dividing total processes into small nodes, we can choose the proper computer resources for each process. These resources can range from single-board computers such as Raspberry Pi or the Next Unit of Computing (NUC) for low-level sensor drivers, to workstations for high-level recognition or planning processes.

Furthermore, it is quite easy to install newly developed processing nodes to the system stably to carry out the functions of the ROS, and a number of open resources and packages available worldwide can be easily utilized in ROS-TMS.

Fig. 3 shows an example of the node structure in a room. Pressure sensors and RFID tag readers are installed in a cabinet and a refrigerator, and the positions and names of objects in the cabinet and refrigerator are recorded in the database (TMS_SD) though the sensor nodes (TMS_SD). Laser range finders are installed on the floor, and range data are sent from the sensor nodes (TMS_SD) to the processing nodes (TMS_SA) to recognize the objects and track their movements. The motions of robots, humans, and objects are estimated according to the tracking results. A 9-axis acceleration sensor, an angular acceleration sensor, and a magnetic field sensor are installed in the furniture, such as a wagon or a wheelchair that can move freely. These sensors are used to track each position by combining the tracking information obtained by the laser range finders. RGB-D cameras (Kinect) are attached to the robots and the walls to recognize objects or human behavior.

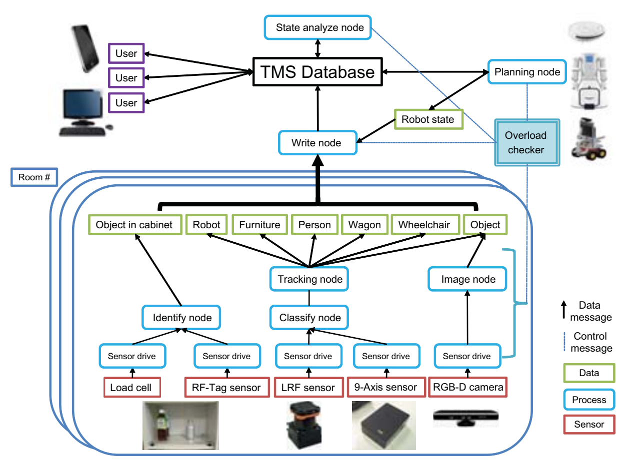

Fig. 3. Processing nodes of TMS

Hierarchical modular structures similar to ROS-TMS can be seen in other robot control architectures and smart house controllers [21] [22] [23] [24] [25] [26] [27] [28] [29]. For example, Fong et al. [26] proposed a middleware called "Human-Robot Interaction Operating System" (HRI/OS) for collaboration of humans and robots, and adopted a similar hierarchical structure consisting of distributed processing modules. Techniques that divide the total process into several small processes as modules, which are connected freely according to the required tasks, are popular ways to improve the flexibility and the efficiency in the development of large-scale systems. The characteristics of the proposed ROS-TMS are as follows: the proposed system is based on ROS as a middleware, the flexibility and the expandability are quite high, and a variety of open resources in the world can be utilized by adopting ROS as a middleware.

III. Hardware platform Big Sensor Box

We developed a hardware platform for an informationally structured environment named Big Sensor Box for demonstrating the performance of the software platform ROS-TMS in the daily life environment. The environment, shown Fig. 4, consists of a bedroom, a dining room, and a kitchen.

Fig. 4. Big Sensor Box

In this platform, several sensors, robots, intelligent furniture, and electric appliances are installed. These include the following.

Sensors

- Optical tracking system (Vicon Bonita)
- Distributed RGB-D camera system (Kinect, Xtion)
- Intelligent electric appliances

Service robots

- Humanoid-type service robot (SmartPal V)
- Mobile robot (KXP, Kobuki)
- Wheelchair robot (Mimamoru-kun)

We introduce some sensors, intelligent furniture, and service robots in Big Sensor Box in the following sections.

A. Sensors

1) Optical tracking system: In Big Sensor Box, 18 infrared cameras (Bonita, Vicon Motion Systems) are installed and the three-dimensional positions of infrared markers are tracked. By attaching markers on robots or furniture, as shown in Fig. 5, not only the human motion but also the motion of a robot and the position of furniture are continuously measured with a high level of accuracy that is under one millimeter.

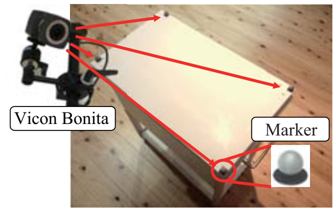

Fig. 5. Optical tracking system (Vicon Bonita) and pose measurement of a wagon

2) Distributed RGB-D camera system: We install nine RGB-D cameras (Kinect v2) in Big Sensor Box and capture the behavior of a human in the room. The position of each camera is calibrated by using a large checker pattern and the optical tracking system mentioned above. Depth images taken by multiple RGB-D cameras are processed by Kinect for Windows SDK 2.0, and the positions of the body and joints of the human are estimated individually in each depth image. Finally, redundant information is fused according to the reliability, which is proportional to the relative angle between the camera's optical axis and body orientation. Only the joint information with the highest reliability is selected as the joint position and stored in the TMS_DB module. Fig. 6 shows the estimated behavior of a human in the room using multiple RGB-D cameras.

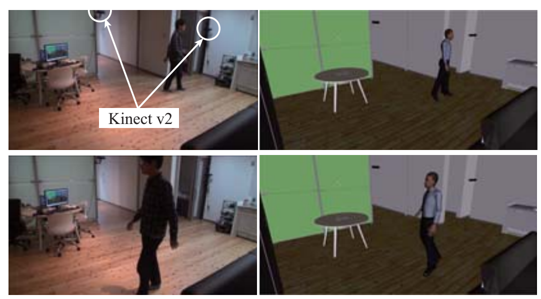

Fig. 6. Motion estimation using RGB-D cameras (Kinect v2)

3) Intelligent furniture: In the cabinet and the refrigerator in Big Sensor Box, several load cells and RFID tag readers are installed, and the positions and names of objects can be detected as shown in Fig. 7.

The bottom of each object in Big Sensor Box has an attached RFID tag. The position of the object placed on the cabinet or the refrigerator is calculated by using the weight distribution measured by the load cells, and the name of the object is read by the RFID tag reader. Measured object information is automatically stored in the database through the ROS_DB module. Thus, by accessing the database, we can check objects in the refrigerator from outside the room (for example, while shopping in a market). In addition, AR markers are attached on each item of furniture, such as the refrigerator. By capturing images by using smart phones or smart glasses, we can know the objects inside without opening the door. Fig. 7 shows how we check the registered objects in the intelligent refrigerator through a smart phone.

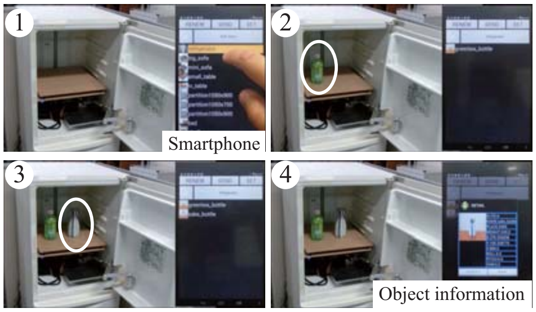

Fig. 7. Registration of objects in intelligent refrigerator. 1. Select refrigerator, 2. Plastic bottle is detected, 3. Ceramic bottle is detected, 4. Information of ceramic bottle is displayed.

B. Service robots

1) Humanoid-type service robot: Fig. 8 shows a humanoid-type service robot (SmartPal V, Yaskawa Electric). On top of the robot, an omnidirectional laser scanner (Velodyne HDL-32E) and an RGB-D camera (Xtion, ASUS) are attached, and so the robot can avoid collisions with an environment, human, and object in real time, as shown in Fig. 8.

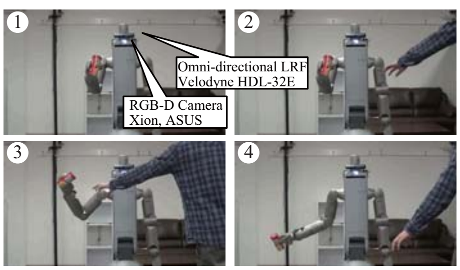

Fig. 8. 1. Humanoid-type service robot (SmartPal V, Yaskawa Electric Corporation). 2-4. The robot avoids to hit a human hand using Velodyne HDL-32e, thus tries to keep holding an object by avoiding a human hand.

2) Mobile robots: Fig. 9(a) shows a small service robot with a manipulator (Katana, Neuronics), an RGB-D camera (Xtion, ASUS), and a mobile platform (Pioneer3-AT, MobileRobots) named KXP. Katana is a 5-DoF manipulator with a maximum capacity of 0.5 Kg and the ability to grasp an object on a floor by using a 1-DoF gripper hand. Fig. 9(b) is the “kobuki”, which is a standard platform in ROS. An RGB-D camera is mounted on the top of the kobuki.

3) Wheelchair robot: We developed a wheelchair robot for transportation assistance and vital data acquisition, as shown in Fig. 9(c). The user wears vital sensors such as an electroencephalography (EEG) sensor (MindWave Mobile, NeuroSky) and a heart rate sensor (T31C transmitter, POLAR). The acquired vital data are transmitted to ROS-TMS via Wireless LAN/Bluetooth and stored in the database. A doctor or a nurse can access these data remotely by using a smartphone or a tablet to monitor the health status of the user. With this system, ROS-TMS can manage not only the information of robots or objects but also vital data of the human.

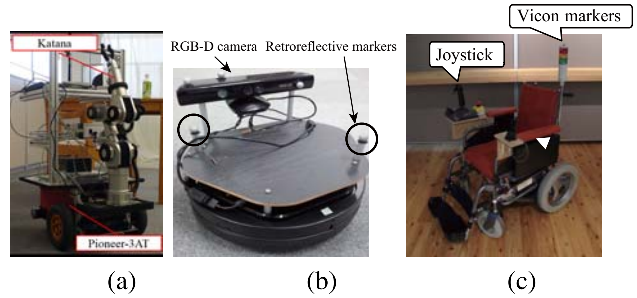

Fig. 9. Service robot (KXP) and wheelchair robot

Furthermore, the position of the wheelchair robot in Big Sensor Box is tracked by the optical tracking system described above and stored in the database. If the user taps a tablet to select the destination on a displayed map, the wheelchair robot can move to the destination automatically.

C. Simulators

Several simulators (Choreonoid (AIST), RViz, and Gazebo) are currently available for Big Sensor Box. Fig. 10 shows an example of the Choreonoid simulator. Fig. 11 shows an example of Big Sensor Box reproduced by RViz and Gazebo.

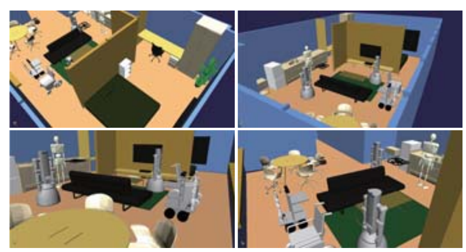

Fig. 10. 3D model of ”Big Sensor Box” (Choreonoid)

Fig. 11. 3D model of ”Big Sensor Box” (RViz and Gazebo)

IV. Robot Service Experiments Using ROS-TMS and Big Sensor Box

We carried out robot service experiments in Big Sensor Box. The aim of these experiments is to confirm that the developed software platform ROS-TMS 4.0 and the hardware platform Big Sensor Box have competent ability and performance capable of providing some typical robot services. Especially in this paper, we focus on two service tasks in daily life, fetch and give task and automatic transportation task by wheelchair robot. Both of them are frequent activities of daily life in care homes. More concretely, we confirm the following.

1) Can robots with different structures provide the same fetch and give task by referring to the database?
2) Can the wheelchair robot execute the automatic transportation task? Can the user move to the desired location automatically by tapping the map on a tablet?

In the fetch and give task, the user does not specify the exact location of the object. Instead, the user just gives the name of the object to ROS-TMS, and ROS-TMS specifies the position of the object by referring to the database in which the information of the objects, the robot, and the user is registered beforehand. The ROS-TMS plans a secure robot trajectory toward the object and a feasible grasping motion, then executes “move”, “grasp”, “move”, and “give” subtasks according to the task execution machine, SMACH [30].

A. *Fetch and give tasks by two types of service robots*

1) Execution procedures for fetch and give task: We conducted experiments for the fetch and give task by using two types of service robots. In this task, the robot moves to the position of the object desired by the user, grasps the object, approaches the user, and passes the object to the user.

All the procedures are planned and controlled by ROS-TMS. First, the task command received through the TMS_UR module is divided into several subtasks, “move”, ” grasp”, and “give”, according to the task description in the database. Each subtask is sent to the TMS_RP module sequentially by SMACH, shown on the left side of Fig. 12. The TMS_RP module produces detailed motion instructions, such as the trajectory of movements or grasping motions by interpolating the object position, the cabinet position in which the object is placed, the current robot status, and the user position stored in TMS_DB. The motion instructions produced in the TMS_RP module are sent to the RMS_RC module, translated to the actual motion patterns by considering the structure of each robot, and executed by each robot. Due to this architecture, the same motion instruction sets (a sequence of subtasks) for a desired task can be repeatedly used, even by robots with different structures. In addition, same subtasks can be reused for other tasks repeatedly.

Here we explain the procedures for task executions in more detail. Tasks, subtasks, their parameters, and execution orders are stored in the database as a sequence of characters, as shown in Table I. For example, get_object task (ID=8001), which is a fetch and give task, is composed of the move subtask (ID=9001), the grasp subtask (ID=9002), and the give subtask (ID=9003). In addition, each subtask is called with an object ID ($oid), user ID ($uid), and robot ID ($rid). For the description of subtasks, we adopted the inverse Polish notation and defined a delimiter with “(space)”, and serial and parallel connections with “+” and “|”, respectively. Each object ID ($oid) or user ID ($uid) is translated to actual coordinates of the positions stored in the database during the task execution. When tasks such as “get_object” are called, the robot ID ($rid) must be set for a robot (2020: SmartPal, 2003: KXP, etc.) according to the user requests to perform the service tasks.

TABLE I
TASK DESCRIPTIONS IN TMS_DB

Type	ID	Name	Code
Task	8001	get_object	9001$oid 9002$oid + 9003$uid +
Task	8002	patrol	9001$rid 9006$oid 9007$oid | +

Type	ID	Name
Subtask	9001	move
Subtask	9002	grasp
Subtask	9003	give
Subtask	9006	sensing
Subtask	9007	random_walk

In the case that $rid = 2002 (SmartPalV), $oid = 7001 (snack), and $uid = 1001 (the user), the input scripts about the get_object task for SMACH issued by TMS_TS are as follows:

```
smach.StateMachine.add('move',
ServiceState('rp_cmd',
rp_cmd,
request = rp_cmdRequest(9001,2002,[7001])),
transitions={'succeeded':'grasp'})

smach.StateMachine.add('grasp',
ServiceState('rp_cmd',
rp_cmd,
request = rp_cmdRequest(9002,2002,[7001])),
transitions={'succeeded':'give'})

smach.StateMachine.add('give',
ServiceState('rp_cmd',
rp_cmd,
request = rp_cmdRequest(9003,2002,[1001])),
transitions={'succeeded':'succeeded',
aborted':'aborted'})
```

2) Experimental results: Before the experiments using the service robots, we confirmed that the actions produced by ROS-TMS were safely executable by the Choreonoid simulator. Fig. 12 shows the sequence of actions planned for the fetch and give task explained in Section IV-A.1. Fig. 12 shows the planned actions for each subtask. The dotted lines on the floor show the desired trajectories planned by TMS_RP. These trajectories are so-called Voronoi edges, which are the most distant lines from obstacles, and the robot moves on these lines safely. The diagram on the left side of each figure shows the current execution status of subtasks managed by SMACH.

Next, we conducted actual experiments using two types of service robots, SmartPAL V and KXP. Fig. 13 shows that SmartPAL V and KXP are performing the desired service task with the same task descriptions issued by TMS_RP. The structure differences between the two robots are negated by TMS_RC. We can see that the user can receive the desired object placed anywhere in Big Sensor Box if the object position is registered in ROS-TMS using the intelligent cabinet/refrigerator or the optical tracking system. If we call the same task description by changing robot/user/object IDs, ROS-TMS translates these IDs to the actual positions by referring to the database and performs the desired task appropriately.

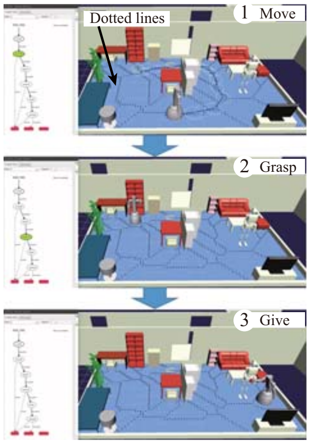

Fig. 12. Fetch and carry task performed by service robot (Simulation). 1. Move to a shelf, 2. Grasp an object, 3. Move and give an object.

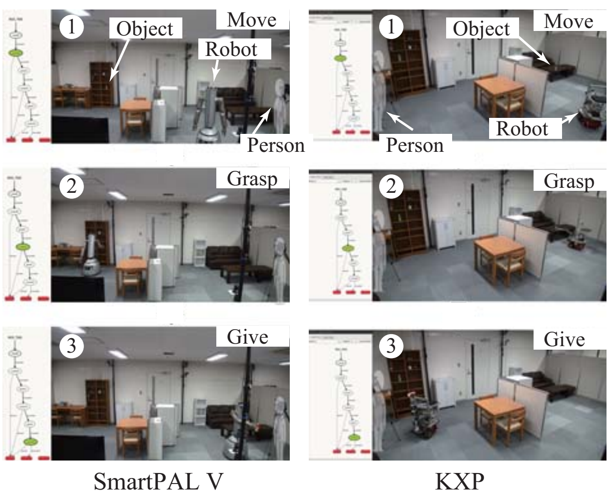

Fig. 13. Fetch and carry task performed by service robots (SmartPAL V and KXP). 1. Move to a shelf, 2. Grasp an object, 3. Move and give an object.

Fig. 14 shows the demonstration using the voice input node in TMS_UI. The user asks the system by voice, for example, "SmartPal, bring me a snack". Then the TMS_UI transforms the voice input to the corresponding task description and parameters, and TMS_RP and TMS_RC execute the fetch and give task automatically through SmartPAL V. The left figures of Fig. 14 show the robot motion and the right figures show the planned motions displayed in RViz.

The robot position is measured by the markers attached on the body and the optical tracking system. The user position is measured by using markers on glasses worn by the user. The positions of objects and furniture are registered in the database beforehand. The error and the sampling period of the optical tracking system are less than 1 mm and 100 Hz, respectively, and thus the robot can move and grasp an object without collisions with the environments or objects in the environments.

Through the experiments, we confirmed that the fetch and give task can be performed with different types of service robots, sensors, user interfaces, and simulators by the software platform ROS-TMS and the hardware platform Big Sensor Box.

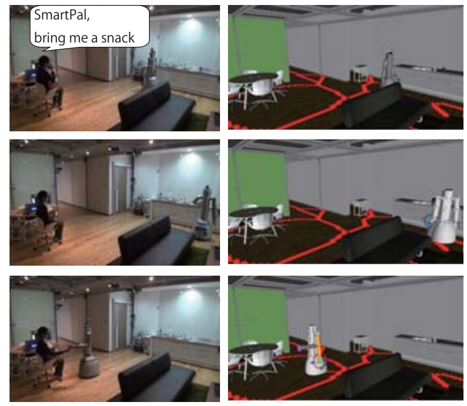

Fig. 14. Fetch and carry task performed by service robot in Big Sensor Box according to voice input (Simulation and experiment)

Fig. 15 shows an example of a "patrol task" performed by the kobuki. The patrol task is a service task that moves randomly and searches for a fallen person in the room. This task consists of subtasks "Move" (move to the center of the room), "Random walk" (move randomly), and "Sensing" (search for a fallen person by using an RGB-D camera), and includes the same subtask (Move) used in the fetch and give task. The subtasks "Random walk" and "Sensing" are executed in parallel by SMACH. When a large object with a size similar to that of a human is detected, the task is terminated with the abort instruction, the robot stops at the current position, and beeps to notify the abnormal detection. This example suggests that the same subtasks can be utilized repeatedly in several tasks, because the proposed system has high reusability and the flexibility.

Fig. 15. Patrol task performed by the kobuki

B. Automatic transportation task using a wheelchair robot

One of the key applications of the proposed platforms ROS-TMS and Big Sensor Box is elderly care service in a care home. Therefore, we conducted experiments for an automatic transportation task using the wheelchair robot we developed. The aim of this experiment is to confirm the following points:

1) A patient can move automatically and safely to the desired location.

2) Vital data of a patient can be measured and stored in the database so that a care giver can monitor the health condition.

The scenario in this experiment is as follows: a patient moves from the sofa to the table, takes a plastic bottle, and moves to the bed. More concretely, this scenario consists of the following sequences.

1) A patient calls the wheelchair robot by tapping the tablet, then the wheelchair robot comes to the sofa automatically.
2) The patient transfers to the wheelchair robot.
3) The patient moves to the desk manually by using a joystick on the armrest.
4) After taking a plastic bottle, the patient taps the bed image displayed on the tablet and the wheelchair robot transports him to the bed automatically.

Figs. 16 and 17 show the experimental results and the target position of the wheelchair robot displayed on the tablet, respectively. Fig. 18 shows the vital data, which are heart rate (hr), meditation level (med), and attention level (att) estimated by the electroencephalography (EEG) sensor. These data are displayed on the Choreonoid simulator.

Through the experiments, it is confirmed that a patient can move anywhere in Big Sensor Box automatically and manually by the wheelchair robot while the vital data is measured, recorded, and monitored by the care giver.

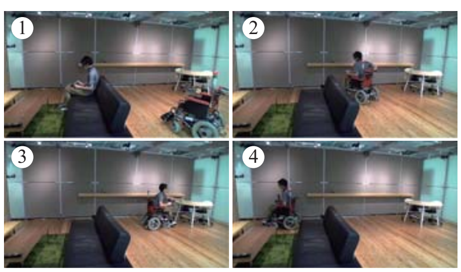

Fig. 16. Service experiment performed by wheelchair robot. 1. Call a wheelchair robot by a tablet, 2. Get on a wheelchair robot, 3. Move manually, 4. Move automatiaclly.

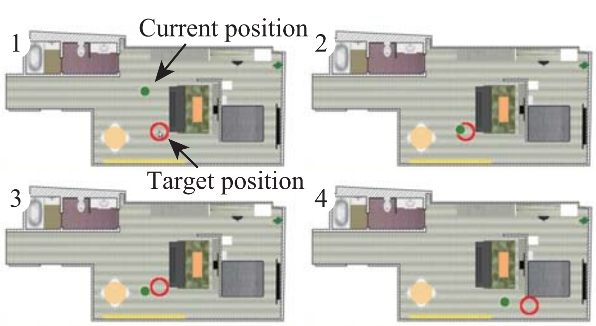

Fig. 17. GUI images for service experiment. 1. Call a wheelchair robot by a tablet, 2. Get on a wheelchair robot, 3. Move manually, 4. Move automatiaclly.

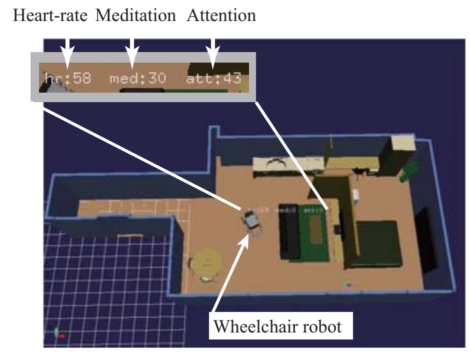

Fig. 18. Vital data displayed on Choreonoid

V. CONCLUSIONS

In this paper, we introduced the software platform ROS-TMS for an informationally structured environment. In addition, we proposed the hardware platform Big Sensor Box for the informationally structured environment. Moreover, the hierarchical structure of ROS-TMS and the execution procedure of service tasks in Big Sensor Box were explained with some examples.

Two kinds of service tasks, fetch and give task and automatic transportation task by wheelchair robot, which are activities occurring frequently in daily life, were demonstrated by using ROS-TMS and Big Sensor Box.

One of the main contributions of this study is that we built a house-sized IoT environment based on ROS and showed the feasibility of ROS-based sensor/robot control architectures for realizing a CPS.

We plan to apply the proposed ROS-TMS and Big Sensor Box to elderly care homes in the near future and to confirm the practicality of the proposed platforms.

ACKNOWLEDGMENT

This research is supported by The Japan Science and Technology Agency (JST) through its “Center of Innovation Science and Technology based Radical Innovation and Entrepreneurship Program (COI Program).”

REFERENCES

[1] T. Sato, Y. Nishida, and H. Mizoguchi, “Robotic room: Symbiosis with human through behavior media,” *Robotics and Autonomous Systems*, vol. 18, no. 1-2, pp. 185–194, 1996.
[2] J.-H. Lee, N. Ando, and H. Hashimoto, “Design policy of intelligent space,” in *Proceedings on 1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 1077–1082 vol.3, 1999.
[3] A. P. Pentland, “Smart rooms,” *Scientific American*, vol. 274, no. 4, pp. 54–62, 1996.
[4] R. A. Brooks, “The intelligent room project,” in *Proceedings of the 2nd International Conference on Cognitive Technology (CT ’97)*, CT ’97, (Washington, DC, USA), pp. 271–, IEEE Computer Society, 1997.
[5] J. A. Kientz, S. N. Patel, B. Jones, E. Price, E. D. Mynatt, and G. D. Abowd, “The georgia tech aware home,” in *CHI ’08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’08, (New York, NY, USA), pp. 3675–3680, ACM, 2008.
[6] S. Sugano and Y. Shirai, “Robot design and environment design - waseda robot-house project,” in *Proceedings of International Joint Conference SICE-ICASE, 2006*, pp. I–31–I–34, Oct 2006.

[7] H. Noguchi, T. Mori, and T. Sato, "Automatic generation and connection of program components based on rdf sensor description in network middleware," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2008–2014, 2006.
[8] K.-H. Park, Z. Bien, J.-J. Lee, B. K. Kim, J.-T. Lim, J.-O. Kim, H. Lee, D. H. Stefanov, D.-J. Kim, J.-W. Jung, *et al.*, "Robotic smart house to assist people with movement disabilities," *Autonomous Robots*, vol. 22, no. 2, pp. 183–198, 2007.
[9] Y. Kato, T. Izui, Y. Tsuchiya, M. Narita, M. Ueki, Y. Murakawa, and K. Okabayashi, "Rsi-cloud for integrating robot services with internet services," in *Proceedings of IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2158–2163, 2011.
[10] H. Gross, C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, C. Martin, T. Langner, and M. Merten, "I'll keep an eye on you: Home robot companion for elderly people with cognitive impairment," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2481–2488, 2011.
[11] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz, "The roboearth language: Representing and exchanging knowledge about actions, objects, and environments," in *Proceedings of IEEE International Conference on on Robotics and Automation*, pp. 1284–1289, 2012.
[12] `http://www.ros.org/`.
[13] E. Aarts and R. Wichert, *Ambient intelligence*. Springer, 2009.
[14] `http://www.companionable.net/`.
[15] M. Chan, D. Esteve, C. Escriba, and E. Campo, "A review of smart homes -present state and future challenges," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55 – 81, 2008.
[16] D. Stefanov, Z. Bien, and W.-C. Bang, "The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, pp. 228–250, June 2004.
[17] K. Ashton, "That 'internet of things' thing," *RFiD Journal*, vol. 22, no. 7, pp. 97–114, 2009.
[18] NFS-07-504 `https://www.nsf.gov/pubs/2007/nsf07504/nsf07504.htm`.
[19] `https://github.com/irvs/ros_tms/wiki/`.
[20] Y. Pyo, K. Nakashima, S. Kuwahata, R. Kurazume, T. Tsuji, K. Morooka, and T. Hasegawa, "Service robot system with an informationally structured environment," *Robotics and Autonomous Systems*, vol. 74, no. Part A, pp. 148–165, 2015.
[21] R. G. Simmons, "Structured control for autonomous robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 1, pp. 34–43, 1994.
[22] R. Simmons, R. Goodwin, K. Z. Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in *Proceedings of the first international conference on Autonomous agents*, pp. 245–252, ACM, 1997.
[23] R. G. Simmons, R. Goodwin, K. Z. Haigh, S. Koenig, J. O'Sullivan, and M. M. Veloso, "Xavier: Experience with a layered robot architecture," *ACM Sigart Bulletin*, vol. 8, no. 1-4, pp. 22–33, 1997.
[24] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *The International Journal of Robotics Research*, vol. 17, no. 4, pp. 315–337, 1998.
[25] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert, "Multi-robot cooperation in the martha project," *Robotics & Automation Magazine, IEEE*, vol. 5, no. 1, pp. 36–47, 1998.
[26] T. Fong, C. Kunz, L. M. Hiatt, and M. Bugajska, "The human-robot interaction operating system," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 41–48, ACM, 2006.
[27] J.-H. Kim, I.-B. Jeong, I.-W. Park, and K.-H. Lee, "Multi-layer architecture of ubiquitous robot system for integrated services," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 19–28, 2009.
[28] C.-L. Wu, C.-F. Liao, and L.-C. Fu, "Service-oriented smart-home architecture based on osgi and mobile-agent technology," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, pp. 193–205, March 2007.
[29] S. Das, D. Cook, A. Battacharya, I. Heierman, E.O., and T.-Y. Lin, "The role of prediction algorithms in the mavhome smart home architecture," *Wireless Communications, IEEE*, vol. 9, pp. 77–84, Dec 2002.
[30] J. Bohren and S. Cousins, "The smach high-level executive [ros news]," *IEEE Robotics & Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.