A Robotic System Architecture for Interactive Smart Environment

Wonpil Yu, Jae-Yeong Lee, Kyuseo Han, Joo-Chan Sohn, Minsu Jang, YoungJo Cho

u-Robot Research Division, ETRI, Daejeon, Korea

***Abstract*—This paper describes a robotic system architecture to implement an interactive smart environment in our daily environment incorporating robotics technology. The interactive smart environment comprises a variety of technologies to assist humans inside the environment as well as robots that help humans. The advantage of the proposed architecture lies in the fact that the complexity and heterogeneity inherent in implementing an interactive smart environment can be dealt with convenience by categorizing the environment into three functional spaces: physical, semantic, and virtual space. We have implemented a prototype robotic service framework for an indoor remote monitoring application. We describe various elements constituting the application and how the elements work together to provide a meaningful robotic function in a real environment.**

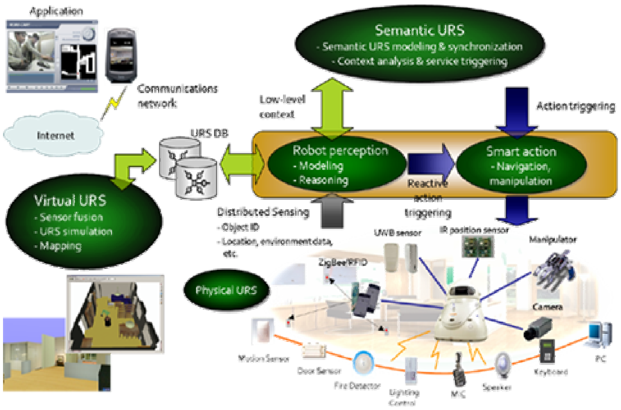


Fig.1 Conceptual structure of a ubiquitous robotic space.

I. INTRODUCTION

THIS paper describes a concerted effort to design and implement an interactive smart environment incorporating robotics in our daily environment. Many researchers have proposed a plurality of robotic system architecture emphasizing synergistic combination of ambient intelligence and networked robotics [1],[2].

Until recently, robots have demonstrated their usefulness in a limited number of applications such as welding, assembly, packaging, and the like. These applications typically comprise a series of predetermined operations in a precisely controlled environment. Today, there is a growing interest for robots to better serve humans beyond manufacturing sites to everyday environments and to accommodate various demands raised as social, industrial, and personal needs change. The above situation renders robot tasks formidable since structuring of environments is subject to restriction and unsupervised understanding thereof is hard to achieve.

While traditional robotics research focused on enhancing functionalities of robots, a new computing paradigm, which is now referred to as *ubiquitous computing*, has been proposed [3]. To date, a number of novel technologies have been proposed supporting the idea of ubiquitous computing: radio frequency identification, wireless sensor network, mobile devices, broadband convergence network, etc.

In the robotics field, many researchers have paid attention to building smart spaces. Although previous works envisioned an intelligent environment [4]-[8], actual environments are still far less intelligent to be characterized by ambient intelligence and networked resources are not dynamically configured yet to meet various task specifications. Implementation and management of a precise localization network in the ordinary environment, running a wireless sensor network taking account of mobility of robots, designing a flexible task manager, and associating a physical environment with a virtual environment for the user interface are some of the reasons that make any smart environment difficult to be realized in the ordinary environment.

In this paper, we describe a robotic system architecture that can be applied to a variety of robotic services, taking account of the above technical issues encountered when we try to implement ambient intelligence together with networked robots into the ordinary environment. To make things easier, we have devised a robotic service framework comprising three conceptual spaces: physical, semantic, and virtual space, which we call ubiquitous robotic space (URS) collectively. Fig. 1 illustrates the conceptual structure of the proposed ubiquitous robotic space.

The proposed URS comprises three spaces: physical, semantic, and virtual space. The current implementation of the physical space consists of a localization network for assisting robot navigation, a wireless sensor network for monitoring environment, and a URS server that manages the physical components. The semantic space has two functionalities of situation understanding and service generation in accordance with the interpretation of the situation. The virtual space provides the user with a way to interact with the physical space by utilizing existing IT infrastructure such as wireless communications network and the Internet.

The remainder of the paper is organized as follows. Section II describes basic two elements of the physical space---localization sensor network and wireless sensor network, which are key components for realizing the mobility of a robot and sensing the situation of the

Authors are with u-Robot Research Division, ETRI, Daejeon, Korea. All the correspondence should be made to ywp@etri.re.kr.

environment. Section III introduces the semantic space and its constituting elements. The virtual space, which handles environment modeling and user interface to control the physical space, is not described in the paper due to limited space. A robot application making use of the proposed architecture of the ubiquitous robotic space is described in Section IV. Section V concludes the paper and introduces further works.

II. PHYSICAL SPACE

A. Localization Sensor Network

Each localization sensor suite comprises two infrared beacon modules attached on the ceiling of a space in question and an image sensor equipped on the top of a mobile robot. This configuration is, in fact, very well recognized in robotics community [9]-[11]. As is evident, several constraints should be satisfied if any localization technique is to be a practical solution to real-world robotic applications, particularly for building an intelligent space incorporating robot navigation. The following are the constraints that are considered to build a localization network:

- Accuracy: for the convenience of developing robot navigation tasks, the positional accuracy should be less than 20 cm in *x*, *y* each, orientation accuracy being less than 5°.
- Repeatability: for a mobile robot to navigate reliably, jitter in the location information should be bounded by 1 cm of positional error variance and 1° of orientation error variance.
- Coverage: for a localization network to be meaningful, its coverage should be unlimited, which means that the localization network must be highly scalable.
- Response time: update frequency for locating a robot should be high enough. It was set to be more than 10 Hz to provide 3-D localization information (x, y, θ).
- Availability: location information about a robot should be provided at any time of the day and any place of an indoor environment.
- Deployment: the localization network should be capable of wireless operation except for power supply.
- Cost: constituting elements should be cost effective for the proposed localization network to be deployed in a large indoor environment.

The localization network plays a key role underlying the whole ubiquitous robotic space. As can be noticed, in order to build a ubiquitous robotic space as a serious business solution based on robotic technology, the constraints above should be completely satisfied. We chose an optical tracking scheme to realize a localization network satisfying the constraints above. Fig. 2 is a basic sensor configuration to build the localization network.

The proposed sensor suite is configured such that infrared beacon modules are attached on the ceiling of a space in question and an image sensor is mounted on the top of a mobile robot as shown in Fig. 2. The image sensor is a CCD camera having an infrared band-pass filter. It is oriented to look upward so that its optical axis is perpendicular to the ground. For the sake of maximal field of view, a wide angle camera lens is utilized. Each beacon module contains an infrared LED whose on/off status is controlled externally by wireless communication. In order to control LEDs of the beacons independently, a unique beacon ID is assigned to each infrared beacon module.

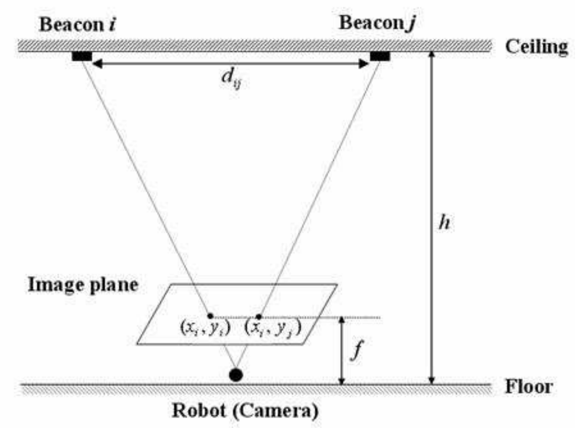


Fig. 2 Sensor configuration for estimating robot location

The localization is performed in two steps: in the first step, the image coordinates of the infrared LEDs are computed and tag IDs are then identified. In the second step, 2-D position and heading angle of a robot are computed from the image coordinates and world coordinates of the detected LEDs.

B. Wireless Sensor Network

The sensor network platform to build a ubiquitous robotic space is called *u-Clips* and composed of sensor node hardware and sensor network protocol software. The u-Clips sensor node is designed to meet common requirements that a wireless sensor network platform should satisfy: small size and low power consumption.

The *u-Clips* sensor network protocol stack is based on ZigBee network protocol [12], which is a de facto standard sensor network protocol due to its support for low power and reliable data communications. The *u-Clips* sensor network protocol stack further includes a mobility-supporting procedure in addition to the ZigBee network protocol to support mobility of a mobile robot equipped with a wireless sensor or sink node.

1) u-Clips Sensor Node

Fig. 3 shows the u-Clips sensor node hardware consisting of three parts: a processor board, sensor board and programming board. The processor board includes a CC2430 main processor, 2.4 GHz PCB antenna, and SMA antenna connector. It also accommodates a couple of basic sensors such as temperature, humidity, light sensor, and magnetic door sensor. The sensor board is piggy-backed on the processor board and provides a dedicated sensing functionality. The programming board provides a USB and RS232C interface for rendering the processor board into a sync node. A processor board is piggybacked onto the programming board, thus being connected to a robot and providing the robot with sensor data collected from each individual sensor node.

2) u-Clips Sensor Network Protocol

a) ZigBee Network Protocol: Since main characteristics of ZigBee meet the requirements imposed on common sensor network applications, ZigBee protocol stack has been widely used in applications such as smart home and building automation. We do not need to construct a network topology manually because the underlying ZigBee stack provides a self-configuration feature. A sensor node that is first turned on forms a network with a designated PAN ID in a specific channel. Succeeding sensor nodes including a mobile robot discover the network and check the PAN ID. If the PAN ID is found to be the one in question, the succeeding sensor nodes join the network automatically.

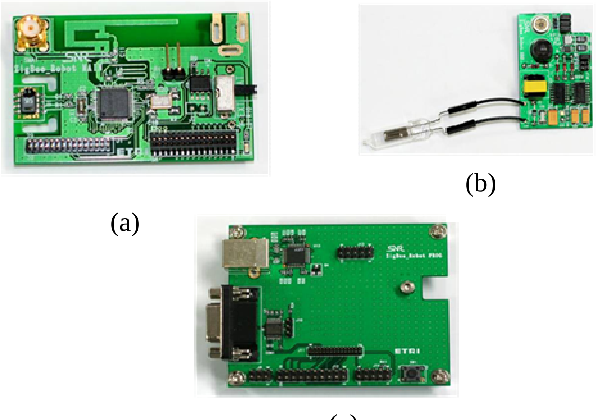

(a) (b) (c)

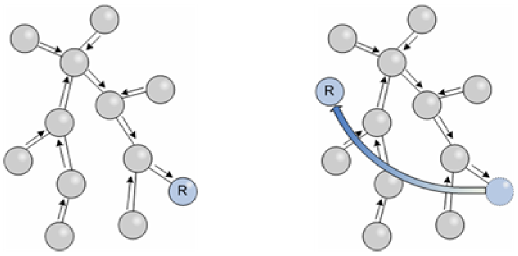
Fig.3 Hardware components of u-Clips sensor node. (a) Processor board. (b) Sensor board. (c) Programming board.

After the above network formation procedure is finished, constituting sensor nodes transmit their sensor data to the robot (sync node) using a tree routing scheme of ZigBee network. Thanks to the addressing scheme of ZigBee, data from sensor nodes can reach the robot by using the tree topology without any routing table as illustrated in Fig. 4-(a). As can be noticed, tree routing requires little memory overhead and can also significantly reduce route discovery overhead compared with a table-driven mesh routing algorithm.

Different from existing sensor network applications, a mobile robot equipped with a wireless sensor or sink node introduces a peculiar feature to a conventional sensor network. If the robot moves out of an RF transmission range of its current parent node, it cannot communicate with the entire sensor network as shown in Fig. 4-(b). However, it is an essential requirement for a sensor network collaborating with robots to guarantee undisrupted communication with the mobile node by managing the mobility thereof.

b) Mobility Supporting Procedure: Currently, ZigBee specification version 1.0 [12] does not support node mobility. To support the mobility of a mobile robot, we propose that a node installed on a mobile robot (which is now referred to as a mobile node) should behave as an end device in a ZigBee network. In other words, a mobile node must not have any child nodes in a sensor network. It is because the communication link between the mobile node and its child nodes are broken as the robot moves. If it is the case, all the data transmission from ancestor nodes of the child nodes will be disabled.

We should also take into account how to announce a new network address of the mobile node to the entire sensor network. Out of transmission range, the mobile node will try to re-establish its communication link to a nearby node at the rejoining phase of the proposed procedure. If the mobile node rejoins the sensor network successfully, it gets a new network address according to the ZigBee specification. In this case, sensor nodes retaining a previous network address of the mobile node cannot communicate with the mobile node.



(a) (b)

Fig.4 (a) Tree routing algorithm. (b) Mobility problem.

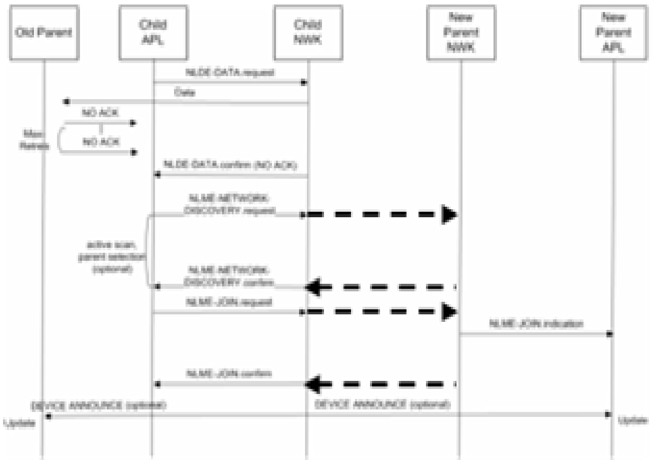


Fig.5 Proposed mobility supporting procedure.

III. THE SEMANTIC SPACE

Figure 7 shows the overall structure of the proposed semantic space. The semantic space carries out two main functions: situation understanding and service decision. For situation understanding, the semantic space accepts sensory data from the physical space and builds up a set of meaningful interpretations collectively called the semantic world model. By consulting the model, the semantic space derives hypotheses about the current situation, which in turn trigger a series of intelligent robotic services.

As for the situation understanding, the semantic space acts as a dynamic configuration manager. In the design of a URS, sensors or devices can participate or leave the physical space as needed. To cope with the data or services provided by the dynamic participation, it is necessary to maintain timely configurations of the space.

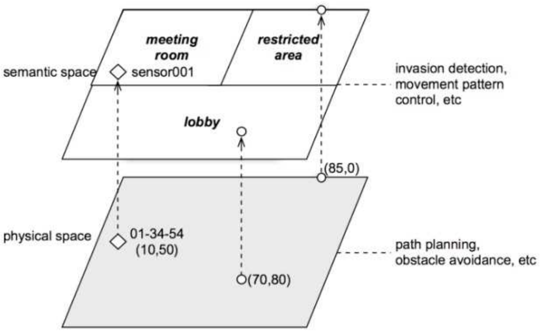


Fig. 6 Coordinates in the physical space are modeled as a symbolic tag in the semantic space, e.g., the coordinate (70, 80) is interpreted as a symbol *lobby*.

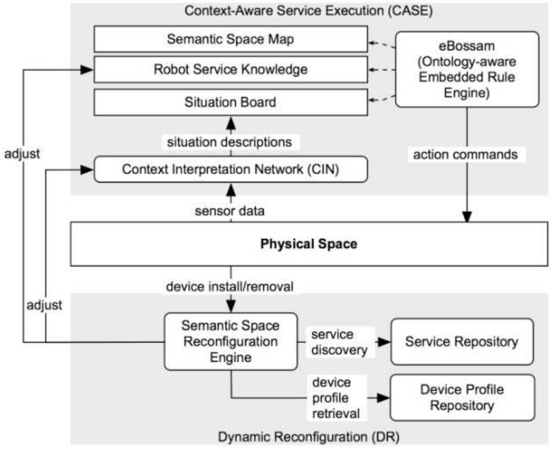


Fig. 7 The semantic space consists of two main function bundles: context-aware service execution (CASE) module and dynamic reconfiguration (DR) module.

Ubiquitous computing environments can be characterized as being distributed, dynamic, and heterogeneous. Exploiting intelligent services in such environments raises not only the issue of modeling and leveraging intelligence but also the issue regarding how various kinds of data and events can be integrated, interpreted, and maintained in a uniform and inter-operable fashion.

We used OWL to model and realize the semantic world model. The semantic space formulates all the accepted data into a set of OWL sentences. Fig. 6 shows how the space is recognized by the semantic space.

The following symbolic sentences represent the device marked as 01-34-54 that is deployed at a position (10, 50) in the physical space.

```
Sensor001
       type Sensor;
       output Temperature;

       positionX 10;
       positionY 50;
       pid "01-34-54".
```

As shown above, 01-34-54 is a temperature sensor and it is deployed in the meeting room. The property pid maps the sensor to the device ID of the physical space. The semantic space, when it receives data originated from an entity identified by 01-34-54, it refers to the above sensor descriptions to interpret the data as a temperature value.

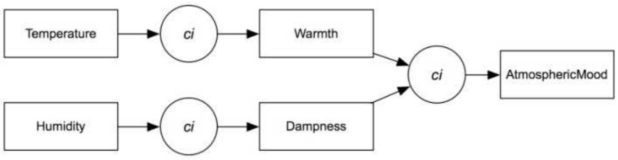


Fig. 8 A sample CIN for deriving atmospheric mood from temperature and humidity.

In what follows, we describe the two main functions of the semantic space in more detail---CASE and dynamic reconfiguration.

A. Context Aware Service Execution (CASE)

CASE module generates service commands based on the semantic world model. The semantic world model consists of the semantic space map and robot service knowledge. The semantic space map is a semantic description of the spatial configuration of the physical space. Each region in the physical space is specified by a unique symbol and a set of semantic descriptors. The following is a sample description to describe a meeting room.

```
region001 type MeetingRoom;
          type RestrictedArea;
          x "0"^^xsd:integer;
          y "45"^^xsd:integer;
          width "30"^^xsd:integer;
          height "30"^^xsd:integer.
```

The description states that `region001` is a meeting room and it is a restricted area. Also, it specifies the regional coordinates that are used to map physical regions into semantic region symbols. Semantic map is used to interpret situations occurring in specific regions or to take actions according to the corresponding region type.

The robot service knowledge contains context models and service rules. Context models contain factual descriptions relevant to the service, e.g., employee records, patrol routes during weekdays for company surveillance service. Service rules specify which actions to take on which situations. Situations are tested at the condition part and the actions are initiated at the conclusion part of the rules. Internally, each condition of the rule is matched to the situational facts contained in the semantic space map and the situation board. The following is a simple rule for intrusion detection.

```
rule intrusion-detection is
if
  EntranceLocation(?ent,?loc)
  and RestrictedArea(?loc)
  and (not actor(?ent,?act) or
       (actor(?ent,?act)
  and not admitted(?loc,?act)))
then
  Intrusion(?ent,?loc);
```

The rule fires an intrusion event if an entrance is detected at a location that is a restricted area, and the actor of the entrance has not been identified or the actor is not admitted to the location. `RestrictedArea` comes from the semantic space map and the facts for admitted come from

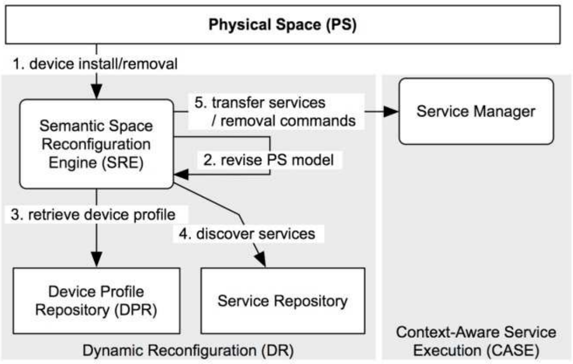


Fig. 9 A schematic flow of operation in dynamic reconfiguration.

robot service knowledge, which are rather static. In contrast, interpreting sensory data from the physical space dynamically creates `EntranceLocation` and actor, for which the context interpretation network (CIN) is responsible.

The CIN comprises a set of hierarchically layered context interpreters. Fig. 8 shows a simple example of a CIN that generates descriptions about atmospheric mood from temperature and humidity. Each circular node, called context interpretation node, is devised with a set of context interpretation rules that are used to perform context conceptualization or synthesis upon its inputs. The rectangular nodes represent what kinds of contexts are produced by the CIN. The nodes at the lowest layer receive raw data from the physical space.

As explained, interpreting context data and deciding the proper service repertoire are executed via logical reasoning on the semantic world model. We use an embedded rule engine called *eBossam*. eBossam, which is developed in C++ language, is a forward chaining reasoner built based on the RETE algorithm [13]. It can recognize URIs and OWL vocabularies as well as the production rules having disjunctions and negation-as-failure in the antecedent, which satisfies the reasoning feature requirements of the current semantic space design.

B. Dynamic Reconfiguration

Dynamic reconfiguration (DR) updates the semantic world model by adjusting the content of the service knowledge and the structure of the CIN. The adjustments change the way the semantic space responds to the situations. Fig. 9 shows the basic flow of operations in dynamic reconfiguration.

The semantic space reconfiguration engine collects all the configuration-oriented events from the physical space and maintains the data model that describes the configuration. The model, called physical space model, contains identification data for all the deployed devices.

DR, whenever it receives a change notification event from the physical space, reflects the changes into the physical space model. If the change event involves a newly deployed device, the physical space provides a unique device identifier, like 01-34-54 as shown in Fig. 6. Based on the identifier, the DR tries to retrieve the device profile by accessing device profile repositories. In our system, we assume that device manufacturers need to make all the basic profiles of their products publicly available on the web.

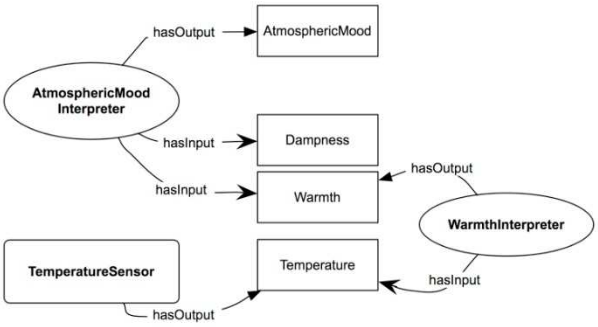


Fig. 10 Sample models of operations in the semantic space.

Once the DR successfully retrieved the profile for the new device, it triggers a service discovery session to search for the services that might be operational in the new configuration of the physical space. If the discovery is successful, it deploys the newly discovered services into the semantic space by transferring the service implementations to the service manager of CASE.

An important issue in dynamic reconfiguration is the mechanism of service discovery. We adopted means-ends planning for discovering and composing services. In means-end planning, operations should first be defined. An operation is defined by specifying its preconditions, inputs, goals, and outputs. We modeled each sensor as an operation that has only the outputs, and the robot service knowledge and the context interpretation nodes as operations having both inputs and outputs. Fig. 10 shows a sample model of operations.

Matching the inputs of an operation and the outputs of another operation accomplishes service discovery. According to Fig. 10, if a newly deployed sensor were identified as a temperature sensor, then a new service called `WarmthInterpreter` would be discovered as a candidate for deployment in the current configuration of the semantic space. Also, if the current configuration has the services for `Dampness` and `Warmth`, the service `AtmosphericMoodInterpreter` becomes deployable. The input and output descriptions for sensors and devices are called device profiles, and those for services, including context interpreters, are called service profiles. As mentioned in the beginning of this section, all of these profiles are specified in OWL, which are served by the corresponding repositories—device profile repository and service repository.

It should be noted that, ideally, by adopting OWL and making the repositories available in the public domain, the semantic space becomes an open platform that makes it possible to deploy any type of devices including robots and disseminating intelligent services.

IV. APPLICATION OF UBIQUITOUS ROBOTIC SPACE

For the evaluation of the feasibility of the proposed ubiquitous robotic space, we have developed a prototype robotic service based on robot navigation. The postulated model is related with a monitoring application, where a robot carries out a routine monitoring task during normal operation. The prototype ubiquitous robotic space was implemented in the ground floor of our building, the area of which was measured to be 22.8 m × 21.6 m.

A sensor network based on ZigBee protocol was also installed in the same space to gather environmental data, which are delivered to the semantic space along with the localization information to infer contextual information. The mean position error of the developed localization network is 4.1 cm with standard deviation of 2.9 cm. Position jitter was measured to be less than 0.5 cm and 0.3°, which well satisfies the repeatability constraint.

The localization network constantly provides location data for the robot and the RSM server, whereby the robot can carries out a navigation task and at the same time, a user may easily checks the location of the robot. A client program was developed for displaying the physical space in a graphics form as well as the raw level status data of the environment in question.

Fig. 11 shows an embodiment of the client program with which the user can monitor and control the proposed ubiquitous robotic space. As shown in the figure, a 2-D metric map is also displayed for inspection of the environment. Irregular situation is detected from the contextual information processing in the semantic space, which subsequently issues a new navigation task to the robot to visit the spot in question. The whole system can be monitored on a remote PC or PDAs; irregular situation is promptly reported to the user through CDMA communications network. By integrating all the elements above, a robot office-monitoring application was experimentally implemented and it was found that the proposed structure of a ubiquitous robotic space provides an efficient framework incorporating various heterogeneous elements into a well orchestrated robotic service platform.

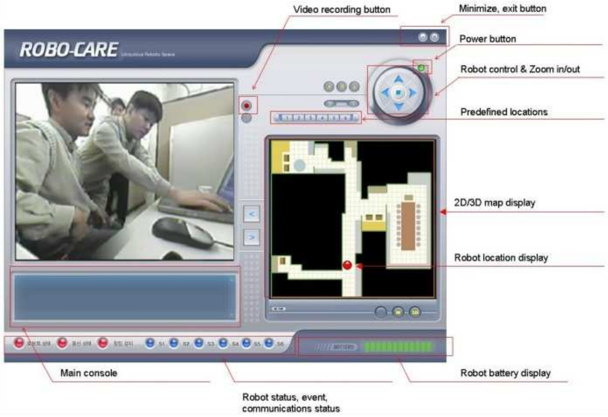


Fig. 11 Implementation of a client program to interface with a ubiquitous robotic space.

V. CONCLUSION

We have described a ubiquitous robotic space, which is flexible robotic system architecture to implement an interactive smart environment. The proposed architecture was implemented in the form of an indoor robotic monitoring application; it can be utilized for a variety of fields, including people monitoring for health care issues, emphasizing the unique feature of networked robots.

ACKNOWLEDGMENT

This work was partly supported by the IT R\&D program of Korea MIC (Ministry of Information and Communication) and IITA (Institute for Information Technology Advancement). [2005-S-092-02, USN-based Ubiquitous Robotic Space Technology Development].

REFERENCES

[1] D. I. Baker, G. T. McKee, and P. S. Schenker, “Network Robotics, a Framework for Dynamic Distributed Architectures,” IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2004, pp. 1768-1771.

[2] M. Broxvall, M. Gritti, A. Saffiotii, B. Seo, and Y. Cho, “PEIS Ecology: Integrating Robots into Smart Environments,” IEEE Int. Conf. Robotics and Automation, 2006, pp. 212-218.

[3] M. Weiser, “The computer for the 21st century,” Scientific American, vol. 265, no. 3, pp. 66-75, 1991..

[4] J. Lee and H. Hashimoto, “Controlling mobile robots in distributed intelligent environment,” IEEE/ASME Trans. Mechatronics, vol. 50, no. 5, pp. 890-902, Oct. 2003J. Lee, K. Morioka, N. Ando, and H. Hashimoto, “Cooperation of distributed intelligent sensors in intelligent environment,” IEEE/ASME Trans. Mechatronics, vol. 9, no. 3, pp. 535-543, Sep. 2004.

[5] T. Sato, “Robotic room: Human behavior measurement, behavior accumulation and personal/behavioral adaptation by intelligent environment,” in IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics, Jul. 2003, pp. 20–24.

[6] T. Sato, T. Harada, and T. Mori, “Environment-type robot system “robotic room” featured by behavior media, behavior contents, and behavior adaptation,” IEEE/ASME Trans. Mechatronics, vol. 9, no. 3, pp. 529–534, Sep. 2004.

[7] N. Y. Chong, H. Hongu, K. Ohba, S. Hirai, and K. Tanie, “A distributed knowledge network for real world robot applications,”in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Sendai, Japan, Sep. 2004, pp. 187–192.

[8] S. Sugano and Y. Shirai, “Robot design and environment design: Waseda robot-house project,” in SICE-ICASE Int. Joint Conf, Oct. 2006, pp. 31–34.

[9] W. Lin, S. Jia, T. Abe, and K. Takase, “Localization of mobile robot based on ID tag and WEB camera,” in Proc. IEEE Int. Conf. on Robotics, Automation and Mechatronics, Singapore, Dec. 2004, pp. 851–856.

[10] Y. Nagumo and A. Ohyae, “Human following behavior of an autonomous mobile robot using light-emitting device,” in Proc. IEEE Int.Workshop on Robot and Human Interactive Communication, Bordeaux-Paris, Sept. 2001, pp. 225–230.

[11] J. S. Park and M. J. Chung, “Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy,” IEEE Trans. Robotics and Automation, vol. 19, no. 2, pp. 250–258, Apr. 2003.

[12] *ZigBee Network Layer Specification 1.0*, ZigBee Alliance, Dec. 2004.

[13] Charles Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", Artificial Intelligence, 19, pp 17-37, 1982