ROBOT BEHAVIOR ARCHITECTURE BASED ON SMART RESOURCE SERVICE EXECUTION

[1]EDUARDO MUNERA, [2]JOSE-LUIS POZA-LUJAN, [3]JUAN-LUIS POSADAS-YAGUE, [4]JOSE SIMO, [5]FRANCISCO BLANES

[1,2,3,4,5]Institute of Control Systems and Industrial Computing, Universitat Politècnica de València, Spain
E-mail: {[1]emunera, [2]jopolu, [3]jposadas, [4]jsimo, [5]pblanes}@ai2.upv.es

Abstract- Robot behavior definition aims to classify and specify the robot tasks execution. Behavior architecture design is crucial for proper robot operation performance. According to this, this work aims to establish a robot behavior architecture based on distributed intelligent services. Therefore, behavior definition is set in a high-level delegating the task execution to distributed services provided by network abstractions characterized as Smart Resources. In order to provide a mechanism to measure the performance of this architecture, an evaluation mechanisms based on a service performance composition is introduced. In order to test this proposal it is designed a real use case implementing the proposed robot behavior architecture on a real navigation task.

Keywords- Robot Behavior, Intelligent Robotics, Smart Resources, and Distributed Systems.

I. INTRODUCTION

Robot task control architectures have been a highly contested issue for years since the first model definition, and formalization, of behavior was exhibited by R. Brooks in [1]. Despite of this it remains as a hot topic, in response to the large number of publications that keep coming on this subject.

This works introduces a robot task organization architecture characterized as an individual behavior. Individual behavior execution relies on distributed services provided by the Smart Resources. As is introduced next, Smart Resources are defined as distributed abstract entities that provides high-level information services, and offer adaptation mechanisms for performance enhancement. As a result, this contribution aims to provide a scalable and extensible architecture, which offers execution flexibility and improves the robot performance.

According to this, next objectives are proposed:

- Review the Smart Resources as service providers and its capabilities.
- Establish behavior architecture based on distributed services access.
- Define a behavior execution evaluation procedure by analyzing the service composition and a behavior execution progress.
- Test and evaluate this proposal on a real robot implementation.

This paper is organized as follows: Related work and similar architectures are reviewed in Section 2. Next, the Smart Resource topology and its main features are detailed. Section 4 describes how Smart Resource promotes the implementation and execution of individual robot behaviors, and how its adaptation mechanisms allow enhancing its performance. In Section 5 is designed a robot behavior execution test to provide a full analysis of this contribution. Finally, some conclusions based on the provided results are addressed.

II. RELATED WORK

As stated on the introduction, the definition of behavior and the establishment of the atomic tasks that composes a behavior execution are common mechanisms for complex robot operation. Classic robot behavior architectures such as, the one introduced in [1] by Brooks or the one presented in [2] by Arkin, aims to characterize and organize the robot execution tasks in a formalized way. These architectures have a strong influence in actual ones. In [3] is introduced an architecture that analyzes the motivational drives of the robot in order to adapt quickly to the to changes in the environment. Some other approaches, like the one presented in [4] proposes an integrated behavior-based control where information about the robot action, is contained in atomic strictures characterized as behavior modules. In [5] is introduced an affordance-based behaviors adapting the task variables to the objective.

In order to define the tasks related with each behavior, some works like [6] offer high-level. Some others like [7], are focused on introducing a toolkit for a full behavior task definition. Solutions like ROSCo [8], or SMACH [9], takes the benefits of a well-known robot software platform like Robot Operative System (ROS) [10] in order to provide a highly integrated behavior definition method.

In order to improve the behavior execution or implementing learning algorithms it is necessary to measure the execution performance of proposed approaches [11]. In [12] is introduced a deploying system that evaluates the robot behavior performance along the execution of domestic tasks. Furthermore, in [13] is studied how a low robot performance can be considered faulty, and whenever that fault can be compensated by collaboration.

Although all these proposals provide good solutions for implementing robot behavior architecture, this work aims to offer a distributed approach in which the behavior definition relies on networked services. This solution enhances the scalability and reusability of the behavior definition, at the same time that abstracts the behavior design from the low level task.

III. PREVIOUS WORK: SMART RESOURCE

Smart Resources, as can be reviewed in Fig. 1, are psychical or cybernetic entities that provide a high-level services which abstracts from device-related matters, data processing tasks and low-level execution architecture. Smart Resources services are developed to be configurable and adapt its execution to fit the service supply requirements. Therefore, Smart Resources have been introduced as a suitable option for a wide range of applications. When dealing with robotics, Smart Resource can provide behavior related services, which defines the execution of required tasks.

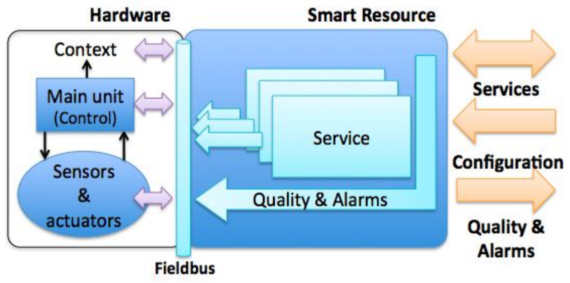

Fig.1. Scheme of a Smart Resource

As in a distributed control systems, robot oriented Smart Resources can provide sensor, actuator and control tasks offered as distributed services. The execution of those services relies on a ROS-based implementation. This integration aims to adapt Smart Resources to provide a ROS compatible interface Smart Resources turns out as ROS-Ready devices [14]. Smart Resource services are accessed through a ROS Multi-peer Architecture (RMPA) communication interface, implemented as detailed in [15]. Main features of RMPA includes are: topic-driven publish-subscriber communication [16], network peer detection and active peer lists management. For each service in the Smart Resource it is provided three different topics related with the configuration, the quality measures, and the service information.

Smart Resources offer the capability of adapting to the system, which is configured to work within some quality bounds. A detailed description of those adaptation mechanisms can be reviewed in [17]. Every time a Service is requested, it must be configured in order to fulfill the need of the client. Main requirements are set in terms of temporal and spatial requirements, information reliability and operation performance.

Robots can perform in a wide range of environments and contexts. Furthermore, mobile robots must face dynamic environments and changing contexts. Although the service quality adaptation mechanism offers an optimum management of Smart Resource capabilities, the context also affects the system efficiency. According to this, the performance of the required services is adapted to the robot context. For this reason Smart Resources can be configured to manage different kind of information according to the context needs, and modify the quality requirements by switching the active System profile.

IV. ROBOT BEHAVIOR ARCHITECTURE

As introduced before, robot behavior is an abstraction layer that is responsible of generating robot commands. Behaviors manage the robot tasks in order to set characterize their execution and the performance requirements. Furthermore, the behaviors are established according to the robot mission and its progress. Robot mission is defined as a sequence of behaviors that are executed in order to achieve a certain goal [18].

Therefore, a behavior architecture based on Smart Resource services execution is addressed along this section. Furthermore, a mechanism for service composition and behavior performance evaluation is introduced.

4.1. Behavior Service Execution

Robot behaviors are characterized in this work as individual behaviors. An individual behavior defines the robot commands for a certain finite action in an exclusive way until its accomplishment. Therefore, only one individual behavior can be active at the same time. Despite of this, in most cases, these individual behaviors are composed by a set of actions of heterogeneous complexity. In order to ease the behavior process definition, these complex behaviors are related with the services provided by Smart Resources. As a result, individual behaviors are characterized as a high level composition of services. Furthermore, the working load of the behavior execution is distributed among the involved Smart Resources. In Fig. 2 is described how a robot defines a set of individual behaviors which accesses the services provided by the Smart Resources in order to delegate complex tasks execution.

In this approach, behaviors configure distributed services to provide the execution support to perform requested tasks. As previously stated, Smart Resource for robotics offers sensor, control, actuator services. By integrating robot-oriented services behaviors can define control tasks in a high level. Nevertheless, both robot control and general-purpose services can be requested by a same individual behavior in order to be integrated into the behaviors structure.

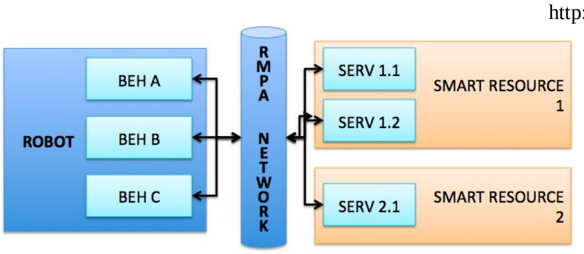

Fig.2. Behavior Architecture accesses the services provided by the Smart Resources in the network.

Behavior execution allows robot to progress on its goal achievement, which is defined as a robot mission. Since the goal of a behavior is to guarantee the robot execution progress, it must interact with the provided services in order to generate robot inputs and outputs that involve environment interaction tasks. These interactions are described as follows:

- Behavior Output: It is designed in order to provide a stimulus according to different kind of requirements. In general case, output information is defined as high-level robot actions that aims to modify its surrounding. To grab/place an object, or to perform a displacement around the environment are some examples. Nevertheless, output can be also characterize as a supply of information, such as displaying information on a screen, play a message through a speaker, or sending a network message.

- Behavior Input: Input signal provide mission-relevant information to the robot behavior. Those inputs are characterized as high-level information which enables the behavior to progress and trigger events such as individual behavior switch or execution accomplishment. Robot environment position, surrounding objects recognition, or network message reception are some of the most common behavior inputs. Furthermore, the quality measures and alarms provided by every requested service are also characterized as behavior inputs.

How to deal with multiple services, how to characterize the influence of the inputs and outputs on the behavior progress, and how it affects to the robot goal accomplishment is addressed on next section.

4.2. Performance and Service Composition

Individual behavior has been characterized to describe robot tasks as high-level composition of multiple services provided by the available Smart Resources. Those Smart Resources not only provide the requested services, but also characterize its execution trough a set of quality measures. This quality measures has been managed by the Smart Resource to provide adaptation mechanisms in order to fit the system requirements specified in the configuration. The Smart Resource services are configured according to the individual behavior requirements. Even when not service adaptation is required, quality measures can provide useful information about the service performance within the behavior context. Those performance values, in addition to other measures, can be used to compute the global behavior performance.

In order to characterize the global performance of the behavior execution, it are evaluated the next parameters: the service performance evaluation, the composed service performance, and the behavior progress factor.

- Service Performance (SP) Evaluation: This evaluation provides information about the service performance. First of all, this measure indicates if the service is running or not. Whenever the service is running, a performance measure is provided according to the average service qualities. Therefore, the contribution of a certain service to the behavior progress can be rated by analyzing this service evaluation measure. Services within the expected performance values will promote a successful execution and progress of the behavior, while poor evaluations warns about a possible stuck or decrement of the behavior progress.

- Services Performance Composition (SPC) Function: Evaluates the performance of the active services in order to compose a global performance value for the whole behavior. The service performance composition function is defined as shown on equation (1). Given a service i the composed performance P_i is computed according to the performance evaluation of all its related services. The relation factor between services is characterized by a weight parameter as expressed on the Service Composition (SC) matrix. Composition weight parameter $sc_{i,j}$ is ranged as [0-1] and define the ratio of dependency between services i and j, where 0 implies no relation and 1 a full dependence between processes. The sum of every row and column in SC must be always 1.

$$SPC = \sum_{i=1}^{n} \frac{\sum_{j=1}^{n} SP_j sc_{i,j}}{n} \quad SC = \begin{bmatrix} sc_{1,1} & \cdots & sc_{1,n} \\ \vdots & \ddots & \vdots \\ sc_{n,1} & \cdots & sc_{n,n} \end{bmatrix} \quad (1)$$

- Behavior progress (BP) factor: Behaviors has been introduced to define a finite action. Therefore, the progress factor provides a measure about the degree of accomplishment of that required action. This factor is expressed numerically in a [0-1] range, where 0 means that the behavior is not started yet, and 1 express that behavior action has been already accomplished. The progress factor is updated periodically as long the behavior is executed. A constant positive progress of this factor points out proper execution of the behavior. Stuck or irregular evolution of factor warns the system from an erroneous behavior execution or definition, which can lead to a robot operation fail.

According to these parameters, the global evaluation of individual behaviors is evaluated through their progress and the services performance composition, as described in the diagram showed in Fig. 3.

These two factors are managed in order to detail the evolution and performance of the behavior execution. As mentioned before, a good behavior design always aims to provide a maximum behavior progress factor and service composed performance values.

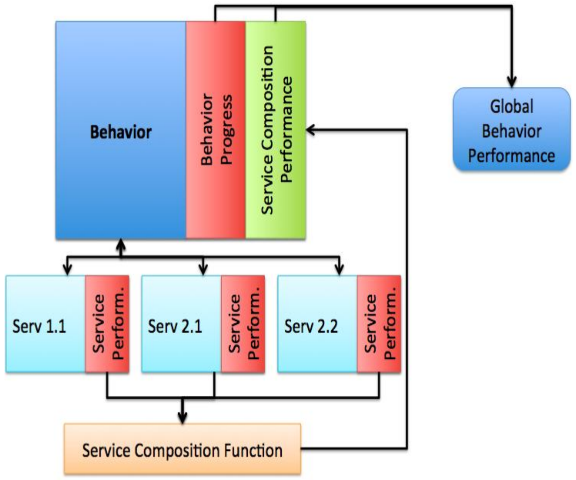

Fig.3. The behavior performance is defined by its own progress and the Service Quality Composition.

The resulting global behavior performance can be used as a measure for monitoring the behavior execution. Furthermore, this value can be used in order to characterize the contribution of the behavior to the robot goal (or mission) accomplishment.

In spite of this, whenever one or several services requires of some external input condition, that can depend on the environment or partner robot, the behavior performance can not be computed. Therefore, these behaviors are defined as 'non-traceable progress' behaviors, and can implement time-out routines for guaranteeing mission progress.

V. RESULTS AND DISCUSSION

In order to test the previously described proposal, it has been designed an experiment which implements the Smart Resources based behavior architecture on a real robot. The chosen robot platform is the Turtlebot II [19]. For these experiments, the Turtlebot Robot has been endowed with a Hokuyo LIDAR sensor in order to be able to scan its surrounding.

Therefor, the current Turtlebot setup, in addition with the available Smart Resources is showed in Fig. 4. Furthermore, the characteristics of these Smart Resources, such as provided services, configuration parameters, and quality measures, can be reviewed in Table 1. According to the current Smart Resources setup there is going to study the implementation of an individual behavior as described in Table 2.

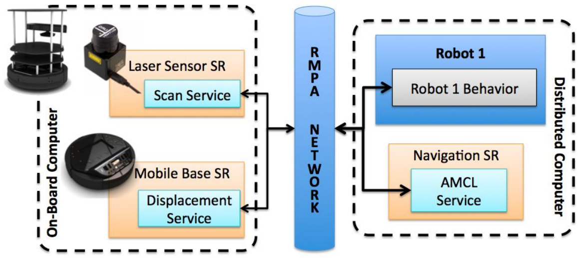

Fig.4. Turtlebot and Smart Resources setup.

Table1: Detail of the Smart Resources.

Smart Resource	Service	Quality Measures	Config. Capabilities	Function
Laser Sensor	Scan	scan_time	resolution	Provides distances from laser scan
Mobile Base	Displacement	Odometry dispersion	speed	Execute robot displacement and compute odometry
Navigation	AMCL	Localization dispersion	goal	Executes a navigation procedure based on augmented montecarlo localization

Table2: Turtlebot behavior definition.

Individual Behavior	Related Services	Description
Navigation	Scan (Laser SR) Displ.(Mob. Base SR) AMCL(Navigation SR)	Sets a navigation goal, establishes a navigation path, and generates displacement commands until the navigation goal position is reached.

Therefore, according to the introduced setup, the Navigation behavior and the required Smart Resources are set as a case of use of individual behavior. How to compute the involved service performance, how are they composed, how to measure the behavior progress factor, and finally how to use those measures to compute the global behavior performance, is going to be detailed.

Given the use case, it is designed an experiment in which the Turtlebot robot is required to achieve a goal position in a previously known environment. For achieving this goal, it has been executed the Navigation behavior, which performance evaluation rate has been set at 1s. The performed displacement of the robot along the map can be reviewed in Fig. 5 and takes around 1 minute to be executed.

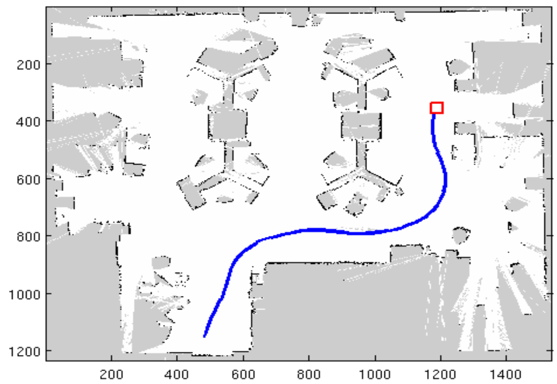

Fig.5. Turtlebot displacement along the map.

During this displacement the Navigation behavior has been executed by accessing to the required services from the available Smart Resources. Therefore, in order to characterize each service performance, along this test it has been stored the service quality measures, previously detailed in Table 1. The statistical information of those values can be reviewed in Fig. 6.

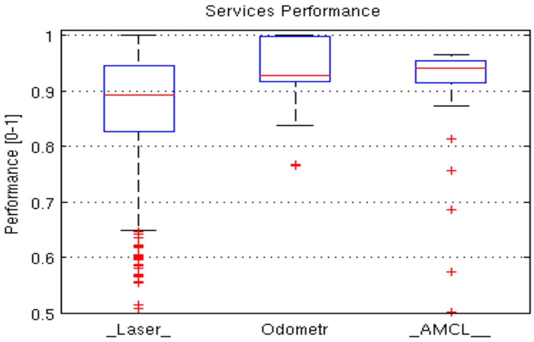

Fig.6. Smart Resource services performance.

Next, those individual service performance evaluations have to be composed according to the equation (1). In order to apply that equation it has been set a Service Composition matrix, which weights the contribution of each service to the final composition. That matrix can be reviewed in Table 3.

Table3: Service Composition matrix.

	Scan 1	Displ. 2	AMCL 3
Scan - 1	0.4	0.1	0.5
Displ. - 2	0.1	0.4	0.5
AMCL - 3	0.5	0.5	0.0

Once the Performance Composition is computed, the behavior progress must be analyzed. As stated before, the Navigation behavior aims to allow the robot to reach a goal position. Therefore, the behavior progress is set as 0 when robot stays at the initial position, and it is set as 1 when the goal position is reached. The behavior progress factor is computed every evaluation step as the relation between the performed displacement and the distance between the start and goal position. The graph in Fig. 7 shows the evolution of the behavior progress and the progress factor along the performed test.

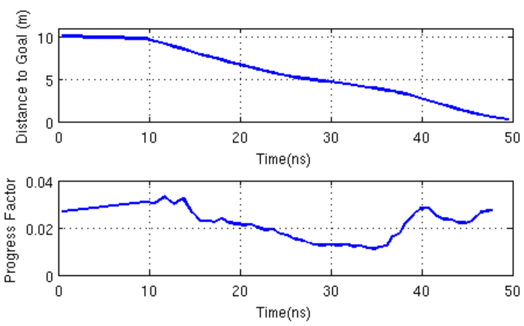

Fig.7. Evolution of the behavior and progress factor.

As has been represented in Fig. 3, once the services performance composition and the behavior progress factor has been computed the global behavior performance is fully characterized.

The statistical analysis of the composed performance allows characterizing the task execution and the suitability of the requested service configuration. A proper service composition design aims to maximize these measures. Despite of this, low values could warns about a mistaken service composition or a to demanding service configuration.

The behavior progress analysis provides information about the execution smoothness. A proper behavior definition provides a low variability of the progress factor long the execution. This is represented as a low statistical dispersion. On the contrary, high dispersion values could warn about an abrupt, or even intermittent, behavior execution, reflecting a low performance that can lead to an execution failure.

According to this, the Navigation behavior is characterized by the results showed in Fig. 8. It can be observed how the mean of the composition measures stay over the 0.9. Furthermore, the dispersion of the behavior progress factor is bounded within a standard deviation of 0.0063.

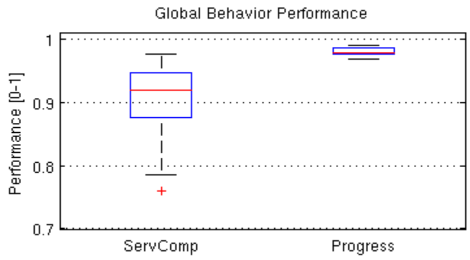

Fig.8. Global Behavior Performance.

As a result, it has been proven that behavior execution can be designed in a high-level layer by relying on a distributed service execution. By implementing this proposal, the behavior execution can be fully characterized through a behavior performance evaluation mechanism. .

CONCLUSIONS

According to the proposal here described, and the results obtained along the performed experiments, major conclusions of this works are as follows:

1. The establishment of service-oriented behavior architecture allows defining high-level robot behaviors. By delegating the task execution problem to the Smart Resources the design, integration, and reusability of the robot behaviors is increased.
2. The development of evaluation mechanisms allows measuring the global behavior performance. By providing a service composition value and a behavior progress factor the robot behavior execution can be fully detailed. Behavior evaluation promotes system performance

enhancements by identifying the most suitable behavior configurations.

3. Along the experiments, the architecture has been proved to perform as expected. Service performance shows a proper task execution, which is reflected in the service composition measures. A constant behavior progress factor has been showed to provide a constant execution until the behavior goal achievement.

ACKNOWLEDGMENTS

Work supported by the Spanish Science and Innovation Ministry MICINN: CI- CYT 866 project M2C2: "Codiseño de sistemas de control con criticidad mixta basado en 867 misiones" TIN2014-56158-C4-4-P and PAID (Polytechnic University of Valencia): 868 UPV-PAID-FPI-2013.

REFERENCES

[1] Rodney Brooks. A robust layered control system for a mobile robot. Robotics and Automation, IEEE Journal of, 2(1):14–23, 1986.

[2] Ronald Arkin. Motor schema-based mobile robot navigation. The International journal of robotics research, 8(4); 92-12. Sage Publications, 1989.

[3] A. Stoytchev and R.C. Arkin. Combining deliberation, reactivity, and motivation in the context of a behavior-based robot architecture. In Computational Intelligence in Robotics and Automation, 2001. Proceedings 2001 IEEE International Symposium on, pages 290–295, 2001.

[4] M Proetzsch, T Luksch, and K Berns. Development of complex robotic systems using the behavior-based control architecture iB2C. Robotics and Autonomous Systems, 58(1):46–67, 2010.

[5] Hermans, T., Rehg, J. M., & Bobick, A. F. (2013, May). Decoupling behavior, perception, and control for autonomous learning of affordances. In Robotics and Automation (ICRA), 2013 IEEE International Conference on (pp. 4989-4996). IEEE.

[6] Vasumathi Raman and Hadas Kress-Gazit. Explaining impossible high-level robot behaviors. IEEE Transactions on Robotics, 29(1):94–104, 2013.

[7] Chien-Ming Huang and Bilge Mutlu. Robot behavior toolkit: generating effective social behaviors for robots. In Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction, pages 25–32. ACM, 2012.

[8] Hai Nguyen, Matei Ciocarlie, Kaijen Hsiao, and Charles C Kemp. Ros commander (rosco): Behavior creation for home robots. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 467– 474. IEEE, 2013.

[9] Jonathan Bohren and Steve Cousins. The smach high-level executive [ros news]. IEEE Robotics & Automation Magazine, 17(4):18–20, 2010.

[10] M Quigley and K Conley. ROS: an open-source Robot Operating System. ICRA workshop on open source software, 3(3.2), 2009. 8, 10

[11] Huang, C. M., & Mutlu, B. (2014, March). Learning-based modeling of multimodal behaviors for humanlike robots. In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction (pp. 57-64). ACM.

[12] Siepmann, F., Ziegler, L., Kortkamp, M., & Wachsmuth, S. (2014). Deploying a modeling framework for reusable robot behavior to enable informed strategies for domestic service robots. Robotics and Autonomous Systems, 62(5), 619-631.

[13] Salem, M., Lakatos, G., Amirabdollahian, F., & Dautenhahn, K. (2015, March). Would you trust a (faulty) robot?: Effects of error, task type and personality on human-robot cooperation and trust. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (pp. 141-148).

[14] Munera, E., Poza-Lujan, J. L., Posadas-Yagüe, J. L., Simó-Ten, J. E., & Blanes, F. (2016). Smart Resource Integration on ROS-Based Systems: Highly Decoupled Resources for a Modular and Scalable Robot Development. In Distributed Computing and Artificial Intelligence, 13th International Conference (pp. 331-338). Springer International Publishing.

[15] Simó-Ten, J.E, Munera, E., Poza-Lujan, J. L., Posadas-Yagüe, Munera, E., & Blanes, F. (2016) CKMultipeer: Connecting Devices Without Caring about the Network. Technical Report. Institute of Control Systems and Industrial Computing.

[16] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. ACM Computing Surveys (CSUR), 35(2):114–131, 2003.

[17] Munera, E., Poza-Luján, J. L., Posadas-Yagüe, J. L., Muñoz, M., & Blanes Noguera, J. F. (2015, November). Poster: Context-aware Adaptation Mechanism for Smart Resources. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (pp. 431-432). ACM.

[18] Arkin, R. C., Collins, T. R., & Endo, Y. (1999, November). Tactical mobile robot mission specification and execution. In Photonics East'99 (pp. 150-163). International Society for Optics and Photonics.

[19] Garage, W. (2011). Turtlebot. Website: http://turtlebot.com/.

★★★