International Journal of Scientific & Engineering Research, Volume 7, Issue 11, November-2016
ISSN 2229-5518

CHALLENGES FACING MIDLLEWARE FOR MOBILE ROBOTS IN SMART ENVIRONMENT

M. A. RABBAH(1), N. RABBAH(2), H. BELHADAOUI(1), M. RIFI(1)

(1)RITM Labortory, ESTC, Hassan II University, BP. 8012, Casablanca, Morocco
(2)Laboratory of Structural Engineering, Intelligent Systems and Electrical Energy, ENSAM, Hassan II University, BP. 20000, Casablanca, Morocco

mrabbah@gmail.com

ABSTRACT
The emergence of Internet of Things (IoT) across the World opens new horizons for intelligent mobile robots, communication and coordination between connected objects and mobile robots is one of main research areas, In this paper, we present challenges facing middleware, in order to develop a Distributed Artificial Intelligence (DAI) in collaborative way, and benefit fully robots from their smart environment.

KEY WORDS
Robotics, Connected Objects, Internet of Things (IoT), Middleware, Artificial Intelligence (AI).

1. Introduction

One of the common problems is to extend mobile robots capabilities especially for the old ones that exist already in market. The evolution of sensors enabled robots to better apprehend environment in which they live; However, many challenges remain to raise, among others: the computational power limit of electronic components, the various hardware and software composing robots, and the diversity of communication protocols.

Very large investments in IoT domain are planned over the next few years [1], and installed connected objects reached 4.9 trillion worldwide by 2015 [2], which present big opportunity for robots, allowing them expanding their perceptions and their actions onto the outside world by developing appropriate methods to communicate with (connected objects, databases, web service), so they will be able to solve more problems and offer more services, such as in the medical domain [3].

Middleware comes here to play an important role by providing the necessary architecture for communication, managing data in distributed context, and allowing independence from the heterogeneity of hardware and operating system,

The rest of the paper is organized as follows. Section 2 we will introduce our motivation and vision on which we will base our approach. Section 3 briefly surveys the related work. Section 4 presents the criteria that we will adopt in our study. Section 5 is an evaluation of some middleware. Section 6 concludes the paper and outlines some possible future work.

2. Motivation and background

Robot's AI was developed until now in an individual way, each research team proposes appropriate solutions in relation to a specific situation. We believe that the future of the development of AI, must be done in a collaborative manner, we must be able to provide researchers and even any interested person, a unified way to contribute to the enrichment of solutions, and on the other hand, allow developers to extend the capabilities of their robots/machine by using these solutions in a transparent manner.

To reach this goal we must keep in mind different types of AI, namely: machine learning (deep learning, predictive analytics), natural language processing, planning scheduling and optimisation, perception (speech, vision…), expert systems…

The solution we propose to encourage such effort synergy is a Collaborative Open Platform for a DAI (COPDAI). Figure 1 shows multi-level that compose our vision:

- The Worldwide AI community: it concerns all researchers that want enrich AI repository and solution.
- The Software suite to define and enrich the AI Solutions: It must provide among other things a common language to define AI solutions and a common way to package solutions and expose these solutions like services.
- The Worldwide AI solutions repository: with all utilities provided by the software suite, the community can enrich the different types of AI.
- The Worldwide knowledge repository: this repository will contain data collected from different local smart environment and robots, allowing large coordination between theses

connected object, and allowing community running simulation in differed time based on collected data, this repository is fed either from the local knowledge repository or directly from the community.

- The local AI solutions repository: this repository load solutions from the Worldwide AI solutions repository, to keep local environment independent if communication breakdown.
- The local knowledge repository: this repository contain all data collected in the smart environment and from robots execution, it is free to developer to allow transfer of these data to a Worldwide knowledge repository, the synchronisation can be in real time or in differed time.
- Robots Developers: represent the team that work in specific mobile robot project in smart environment.

We want notice that they may have several levels between Local repository and worldwide repository, such as city repository or other levels, but it will be the same as the relation explained in this figure.

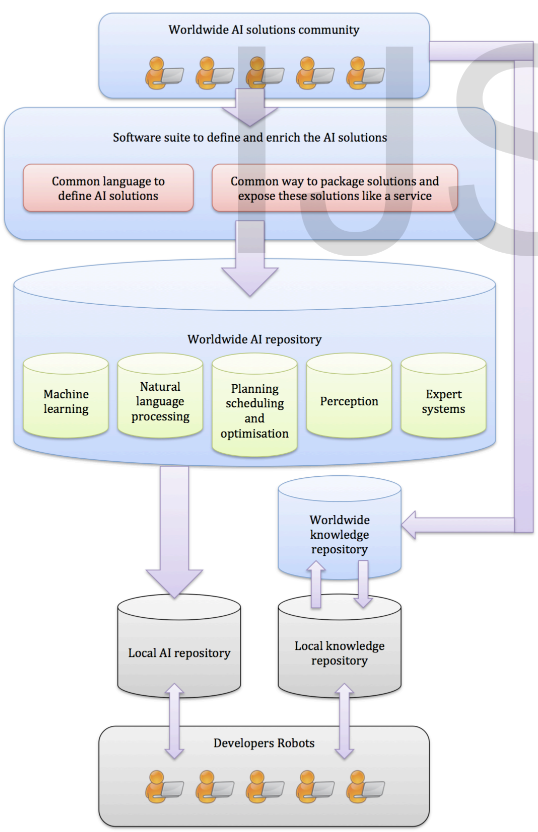

Figure 1. Bird view of our COPDAI vision

To enhance robots perception, they must correctly communicate with the connected objects, and between them. After that came the step to communicate with top level by sending all collected data. The middleware can handle for us this communication, so we were led to make a comparative study, and evaluate some middleware, and their degree of satisfaction to our needs.

3. Related Work

Several studies were made to compare and evaluate middleware; there were some that are focus in middleware oriented smart environment, and other that focus on middleware for robotic applications, here we will make an overview of these studies, lifting the most interesting points that will help us choosing general criteria, and also some of middleware that we will discuss in section 5.

In [4] the authors chose as evaluation criteria: Vendor locking which express the portability of the middleware across multiple platforms, durable data storage services, robustness to failures, management and monitoring middleware components, multi robot coordination services which represent tools to make consensus over network shared values, and at last supported communication. The middleware studied here were Player/Stage [5], Robot operating system (ROS) [6], Miro [7], Microsoft Robotics Developer Studio (MRDS) [8], Marie [9], Orca [10], Pyro [11], DAvinCi [12], Rapytua [13]. According to their study, we decide to include ROS in our evaluation especially because it satisfy partially the durable data storage services and or its open source side, also we retain that none of the presented middlewares are fully suitable for a large scale multi-robot system.

In [14] the authors chose as evaluation criteria: Architectural approach, Robot software used, Simulation Environment, Standards and Technologies Used, Distributed Environment, Security for Controlling Access, Fault Detection and Recovery, Real-Time Capability, Behaviour Coordination, Open Source, and dynamic configuration of connections between services of components at runtime. The middleware studied here were: Orocos [15], Pyro [11], Player/Stage [5], Orca [10], Miro [7], OpenRTMaist [16], ASEBA [17], MARIE [9], RSCA [18], MRDS [8], OPROS [19], CLARAty [20], ROS [6], SmartSoft [21], ERSP [22], Webots [23], RoboFrame [24]. According to their study, we decide to include real time capability and open source as criteria in our evaluation; we also noticed that security still an issue for most of middleware studied including ROS.

In [25] the study targeted middleware for smart environment and smart object, the authors chose as evaluation criteria: Abstraction over heterogeneous input and output hardware devices, Abstraction over hardware and software interfaces, Abstraction over data streams (continuous or discrete data or events) and data types, Abstraction over physicality (location, context), Abstraction over the development process, Heterogeneity and Application Development, Augmentation Variation of Smart Objects, Management of Smart Object, and Evolution of Smart Object Systems. The middleware studied here were: ROS [6], iRoom [26], Aura [27], Context Toolkit [28], JCAF [29], Gaia [30], Ambient Agoras [31], Voyager [32], Smart-Its [33], UbiComp [34], FedNet [35], Smart Products [36], and ACOSO [37]. According to their study, we decide to include ACOSO, JCAF, Voyager, and Aura in our evaluation especially because they are equipped with Knowledge Management Model.

In [38] the study focus on challenges in middleware solutions for the IoT, the authors chose as evaluation criteria: Interoperability, scalability, abstraction provision, spontaneous interaction, unfixed infrastructure, multiplicity, security and privacy. The middleware studied here were: UBIWARE [39], ROS [6], TinyREST [40], and those [41-45]. According to their study, we decide to include UBIWARE as it answer all requirements expect security and privacy.

In [46] the study the adoption of Multi-Agent System (MAS) paradigm, and compared some middleware related to some criteria that we will include in our comparison: Supported operating systems, distributed architecture, communication mechanisms, Naming service, Renaming, Lookup service, Discovery service, Introspection and distributed management tools, Transport mechanisms, Message format and type marshalling. The middleware studied here were: JADE [47], ROS [6], Mobile-C [48], YARP [49], OpenRDK [50], OpenRTM [16], OROCOS [15], ORCA [10].

4. Evaluation Criteria

Before introduction the criteria that we will base on it our comparison, we want raise some important problems that we must take care in order to satisfy our global vision, among other things:

The robots are equipped with their computer system allowing the control of their outputs, and acquire information from their inputs. These systems are either embedded or external or hybrid: on one hand, AI and calculation are external to robot; on the other side, the command part and acquisition are embedded.

Embedded systems in robots run on low performance electronic components, which explain the poor adoption of AI, as it requires very high performance in both calculations and data storage. So can we extend the capabilities of existing robots, without changing their current hardware?

Some Robotic Development Environment (RDE) use latest technologies, in term of programming languages and communication protocols, such as: JAVA, SOAP. However, in real world applications, the limitation of resources make most of these architectures not suitable, on other side, is found more appropriate technologies like C, C++, CORBA but lack when we want extend robots capabilities to better communicate with the outside world [51].

In real life, not all connected object support Ethernet communication, there is a lot of of them that support only Bluetooth or Infrared communication, also we must imagine a solution for those supporting Fieldbus such CANBUS, Serial lines…

After this explanation we can now introduce our evaluation criteria:

- Scalability: Since the IoT is expected to support a large number of devices; scalability seems to be one of the major challenges faced by the middleware approaches.
- Architecture paradigm: each middleware may adopt on or more paradigm like the MAS, Cloud computing, Web services, Component based, Object Oriented, Event based… adopting a paradigm influence radically the way middleware act so it is important to determine this criteria.
- Supported operating systems (OS): the middleware should provide an abstraction over the OS or at least support (Windows and Linux OS).
- Programming language: the middleware should provide an abstraction over the programming language or at least allow developing with main language as C, C++, JAVA, PYTHON, there is other language more suitable for Expert System such as LISP and PROLOG.
- Distributed architecture: We think that a peer to peer (P2P) architecture will bi suitable for communication between connected object and robots, however we think that Client/Server architecture will be more suitable for transmitting data from/to Worldwide knowledge repository or Worldwide AI solutions repository.
- Communication: the communication can be evaluated at different levels:
    1. Protocol of communication: on of the challenge facing middleware is to support multi protocol communication, like: MODBUS, ZIGBEE, Z-WAVE, BLUETOOTH, INFRARED, MQTT… and expose a common way to transmit messages.
    2. Transport mechanisms: most middleware are often designed to transmit message over an Ethernet. However, Field Buses, such as CANBus, I2C, EtherCAT, Serial lines, FireWire, PROFIBUS, and even PCI still a real challenge.
    3. Message format and type marshalling: most known type are XML, YAML, JSON, BINARY… the most important here is the portability of the solution chose by middleware designer.
    4. Node communication: there are different ways that nodes can exchange message such as: Simple messages (broadcast, multicast, send, receive, send/receive), Publish/Subscribe to a certain Topics, Events (They are also

known as the observable/observer patterns), Services (This is a communication mechanism that allows the remote execution of a procedure; the remote procedure call (RPC))…

5. Communication language: language can define not only message form, but also handle-processing information; some attempt was made like here [52] with Horn clause.

- Lookup service: Connected objects or robots will look for a connected object or a robot by services offered, robots and connected objects must register with this directory at boot time, and keep indicating their operational state periodically.
- Renaming: (Remapping). During deployment, a service or a resource must be fully qualified by Universal ID (UID), to avoid any call confusion.
- Naming service: known as the White Pages service, which allows the localization of a resource by name or ID or UID
- Discovery service: A connected object can enter or leave a smart environment at any time, so the middleware must deal with this problem.
- Introspection and distributed management tools: Monitoring, introspection, debugging and administration are fundamental problems in a distributed system
- Knowledge Management Model: middleware must propose a knowledge database management at Robot level, and at Local Knowledge repository level, and at worldwide knowledge repository, in manner that each part still weakly coupled to other.
- AI repository solutions management: depend on hardware Robot capacity, it can download some solutions from local repository and keep those solution at local level, the same for local and worldwide AI repository solutions, in manner that each part still weakly coupled to other, but in some case Robot can delegate treatment to a higher level if it is out of its capacity.
- Real time capability: commanding some part of robots or connected objects, require real time capability especially in industrial applications.
- Availability as Open source platform: Our goal is to develop a Collaborative Open Platform for a DAI so it is normal that the based bricks must be open source.
- Fault Detection and Recovery: There is always the possibility of a fault at runtime. The faults in the robot's node should be detected and localized, and also, the robot should be able to complete its mission or at least to proceed to a safe mode
- Security: the middleware must ensure the security and confidentiality of information transmitted between nodes

5. Existing middleware Architecture

Following the earlier papers, we have already opted for the following middleware: ROS [6], ACOSO [37], JCAF [29], Aura [27], UBIWARE [39], Voyager [32]. Also we found other interesting middleware to complete our comparison like: LMAARS [53], ICARS [54], COROS [55], SOCRADES [56] and those [57-62].

5.1 ROS: an open-source Robot Operating System

ROS is a tiny P2P middleware; it provides multi-lingual support through interface definition language (IDL). It is based on component paradigm. Messages are marshalled to XML-RPC and based on publish/subscribe communication. Released as open source; its community has grown quickly since 2009. Support Linux and partially windows; there is no explicit security layer and not support real time commands.

5.2 An agent-based middleware for cooperating smart objects (ACOSO)

ACOSO is a middleware based on MAS paradigm, and on an event-driven proactive architecture; two different communication models are adopted (message passing and publish/subscribe), ACOSO currently relies on JADE framework and add device management subsystem, it is platform independent. It satisfy the knowledge management through the KBManager, until writing this lines, ACOSO are not available for download. It supports many communication protocols such as IEEE 802.15.4, ZigBee and 6LowPan.

5.3 Distributed interface bits: dynamic dialogue composition from ambient computing resources (Voyager)

Voyager is a middleware based on Component paradigm, it focus on smart objects in a ambient environment, it support the context awareness, and some communication protocol such as Bluetooth/L2CAP, it wasn't build initially for robotic usage, that's why many concept are missed.

5.4 a service infrastructure and programming framework for context-aware applications (JCAF)

JCAF is a middleware based on Distributed object paradigm, by using Java RMI. It is platform independent; messages are transmitted through publish/subscribe mechanism. A Context Service is looked up using the Java RMI Registry and accessed using RMI invocation. Security is implemented using an authentication mechanism based on a digital signature using the Java Authentication and Authorization Service (JAAS).

5.5 Aura: an architectural framework for user mobility in ubiquitous computing environments

Aura is a middleware for smart environment, its aim goal is maximize the use of available resources, and minimize user distraction and drains on user attention, it is based on distributed objects paradigm, communication protocol support JAVA RMI and CORBA, it support knowledge management by a Context Observer component and it is platform independent, no security layer are implemented.

5.6 SMART SEMANTIC MIDDLEWARE FOR THE INTERNET OF THINGS (UBIWARE)

UBIWARE is a P2P middleware based on MAS paradigm, and FIPA communication protocols, it provides distributed resources histories, and the semantic resource discovery. It not supports natively security layer. It has good scalability and interoperability; the project is on stand by since 2013.

5.7 A Layered Middleware Architecture for Automated Robot Services (LMAARS)

This middleware adopt Web service paradigm, it is divided to tree layers: robot service layer, robot application service control layer, and robot application service layer, the authors propose a context aware workflow language to express the relationship of robot services with service flow, marshalling are done through XML messages, and the SOAP protocol is adopted. No security layer is present.

5.8 A Device Abstraction Framework for the Robotic Mediator collaborating with Smart Environments (ICARS)

This middleware is based on Distributed objects paradigm, it provide an IDL that is compiled into JAVA or C++ interfaces. It is platform independent and use JAVA RMI and CORBA protocols. Its main goal is ambient house.

5.9 Transparent Multi-Robot Communication Exchange for Executing Robot Behaviours

This middleware was based on ROS, it provide additional concept, which can be interesting for as: the robot as service, the authors present a way that allow robot asking some tasks to be done without any previous knowledge about who will execute the service.

5.10 Structuring Communications for Mobile Cyber-Physical Systems

This middleware was based on Agents paradigm, it use TDMA protocol on the top of Commercial Off-The-Shelf wireless communication technologies. The authors introduce the concept of Real Time Database, and to satisfy an efficient resource utilisation, he advises sharing Agents states in period of high team interaction, and separate data access from data transmission.

5.11 Integration of service robots in the smart home by means of UPnP: A surveillance robot case study

The authors explain in this study, the utility of using UPnP protocol, which is widely used today in our daily life. This middleware is platform dependant and more work need to be done to reach a mature level.

5.12 A New ROS-based Hybrid Architecture for Heterogeneous Multi-Robot Systems

This middleware is based on ROS; it comes to solve the weakness in distributed architecture, especially uniqueness of ROS master, which mean there will be always a robot master, and the P2P architecture is so broken. The solution proposed is a cluster of Nodes that collaborate together.

5.13 COROS: A Multi-Agent Software Architecture for Cooperative and Autonomous Service Robots

This middleware aims to implement multi-robot applications on top of ROS, this middleware propose some concept like (problem solving, knowledge base, decentralization, fault tolerance), the knowledge base is used only to store tasks execution states, for remote communication it is based on JSON through UDP, one thing that was interesting in this middleware, that each robot carries its proper knowledge, and refresh this base within a specific timeout.

5.14 An Infrastructure for Robotic Applications as Cloud Computing Services

This middleware provide a solution based on Cloud paradigm, with different level, which answer partially our need for the multi-levels knowledge management. It is implemented using JXTA; the communication between nodes is based on P2P, and it is platform independent.

5.15 SOA-based Integration of the Internet of Things in Enterprise Services (SOCRADES)

This middleware is based on Web service paradigm, the communication use publish/subscribe mechanism and the SOA protocol, it is based on JAVA technologies, and it is platform independent

6. Conclusion

A big work was done until know for developing suitable middleware for both robotic applications and smart environment, but several challenges remain to be raised, among other, standardisation on knowledge representation, developing a collaborative DAI, and securing most of studied middleware. Some project are distinguished from others by their maturity, but no one is perfect, to reach our goal about developing COPDAI, many solutions seam to be a good start point, like COROS and [61], combining multiple paradigm such as Cloud computing and MAS also can be a good solution. Future work will detail each brick in our target architecture. Thereafter, we will instantiate a Framework respecting the proposed model. Finally, we will validate this work by making applications with different types of robots and connected objects.

References

[1] Realising the potential of the connected life. Available from: http://www.gsma.com/connectedliving/wp-content/uploads/2012/05/Realising-the-potential-of-the-Connected-Life-GSMA.pdf

[2] http://www.gartner.com/newsroom/id/2905717

[3] S.A.Akashetal, A novel strategy for controlling the movement of a Smart Wheelchair using Internet of Things, in 2014 IEEE Global humanitarian technology Conference-South Asia Satellite, Trivandrum, pp. 154-158, September 2014.

[4] S.G Chitic, J. Ponge, O. Simonin, “Are Middlewares Ready for Multi-robots Systems?” in Simulation, Modeling, and Programming for Autonomous Robots Volume 8810 of the series Lecture Notes in Computer Science pp 279-290, 2014.

[5] Kranz, M., Rusu, R.B., Maldonado, A., Beetz, M., Schmidt, A.: A player/stage system for context-aware intelligent environments (2006)

[6] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “ROS: an open-source Robot Operating System,” ICRA Workshop on Open Source Software, Kobe, Japan, May 2009.

[7] Kraetzschmar, G.K., Utz, H., Sablatno ̈g, S., Enderle, S., Palm, G.: Miro middleware for cooperative robotics. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 411–416. Springer, Heidelberg (2002)

[8] Johns, K., Taylor, T.: Professional Microsoft Robotics Developer Studio. Wrox Press Ltd., Birmingham (2008)

[9] C. Côté, D. Létourneau, C. Raïevsky, Y. Brosseau and F. Michaud: Using MARIE for mobile robot component development and integration (April 2007)

[10] A. Makarenko, A.B., Kaupp, T.: On the benefits of making robotic software frameworks thin. In: POn the Benefits of Making Robotic Software Frameworks Thin IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego CA, USA, October 29-November 02 (2007)

[11] Pyro. Website (2012), http://pyrorobotics.com/?page=PyroModuleIntroduction/

[12] Arumugam, R., Enti, V.R., Bingbing, L., Xiaojun, W., Baskaran, K., Kong, F.F., Kumar, A.S., Meng, K.D., Kit, G.W.: Davinci: A cloud computing framework for service robots. In: 2010 IEEE International Conference on Robotics and Automa- tion (ICRA), pp. 3084–3089. IEEE (2010)

[13] Hunziker, D., Gajamohan, M., Waibel, M., D'Andrea, R.: Rapyuta: The roboearth cloud engine. In: Robotics and Automation, ICRA (2013)

[14] A. Elkady and T. Sobh, ‘Robotics Middleware: A Comprehensive Literature Survey and Attribute-Based Bibliography”, in Journal of Robotics Volume 2012, Article ID 959013, 15 pages doi:10.1155/2012/959013

[15] P. Soetens, RTT: Real-Time Toolkit, 2010, http://www.Oro-cos.org/rtt.

[16] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. K. Yoon, “RT-component object model in RT-middleware— distributed component middleware for RT (Robot Technol- ogy),” in Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA ’05), pp. 457–462, June 2005.

[17] S. Magnenat, V. Longchamp, and F. Mondada, “Aseba, an event-based middleware for distributed robot control,” in Proceedings of the Workshops DVD of International Conference on Intelligent Robots and Systems (IROS ’07), 2007.

[18] J. Yoo, S. Kim, and S. Hong, “The robot software communications architecture (RSCA): QoS-aware middleware for networked service robots,” in Proceedings of the International Joint Conference (SICE-ICASE ’06), pp. 330–335, October 2006.

[19] C. Jang, S. I. Lee, S. W. Jung et al., “Opros: a new component-based robot software platform,” ETRI Journal, vol. 32, no. 5, pp. 646–656, 2010.

[20] I. A. D. Nesnas, “The claraty project: coping with hardware and software heterogeneity,” in Springer Tracts in Advanced Robotics, vol. 30, pp. 31–70, Springer, Berlin, Germany, 2007.

[21] C. Schlegel and R. Woerz, “Software framework smartsoft for implementing sensorimotor systems,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’99), vol. 3, pp. 1610–1616, October 1999.

[22] Ersp 3.1 software development kit, 2010, http://www.evolution.com/products/ersp/.

[23] O. Michel, “Cyberbotics ltd. webots professional mobile robot simulation,” International Journal of Advanced Robotics Systems, vol. 1, pp. 39–42, 2004.

[24] S. Petters, D. Thomas, and O. von Stryk, “Roboframe—a modular software framework for lightweight autonomous robots,” in Proceedings of the Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware of the International Conference on Intelligent Robots and Systems (IEEE/RSJ ’07), San Diego, Calif, USA, 2007.

[25] G.Fortino, A.Guerrieri, W.Russo and C.Savaglio, “Middlewares for Smart Objects and Smart Environments: Overview and Comparison”, in Internet of Things Based on Smart Objects, Internet of Things, DOI: 10.1007/978-3-319-00491-4_1, 2014

[26] Collett, T.H.J., Macdonald, B.A.: Player 2.0: toward a practical robot programming frame- work. In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA) (2005)
[27] Brooks, R.A.: The intelligent room project. In: Proceedings of the 2nd International Con- ference on Cognitive Technology (CT '97). CT '97, pp. 271–278. IEEE Computer Society, Washington, DC, USA (1997).
[28] Sousa, J.P., Garlan, D.: Aura: an architectural framework for user mobility in ubiquitous computing environments. In: Proceedings of the IFIP 17th World Computer Congress—TC2 Stream / 3rd IEEE/IFIP Conference on Software Architecture: System Design, Development and Maintenance. WICSA 3, pp. 29–43. Kluwer, B.V., Deventer, The Netherlands (2002).
[29] Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum. Comput. Interact. 16(2), 97–166 (2001).
[30] Bardram,J.E.: The java context awareness framework (JCAF) — a service infrastructure and programming framework for context-aware applications. In: Proceedings of the Third Inter- national Conference on Pervasive Computing, ser. PERVASIVE'05, pp. 98–115. Springer, Berlin, Heidelberg (2005).
[31] Roman,M.,Campbell,R.H.: Gaia: enabling active spaces. In: Proceedings oft he9thWork- shop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Oper- ating System. EW 9, pp. 229–234. New York, NY, USA, ACM (2000).
[32] Savidis, A., Stephanides, C.: Dynamic environment-adapted mobile interfaces: the voyager toolkit. In: Array (ed.) Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), vol. 4, pp. 489–493. Lawrence Erlbaum Associates, New Jersey (2003)
[33] Beigl,M.,Gellersen,H.W.:Smart-its: an embedded platform for smart objects .In: Proceedings of the Smart Object Conference (SOC 2003), pp. 15–17 (2003)
[34] Goumopoulos, C., Kameas, A.: Smart objects as components of UbiComp applications. In: International Journal of Multimedia and Ubiquitous Engineering. Special Issue on Smart Object Systems, vol. 4, no. 3, pp. 1–20. sERSC Press, Springer (2009).
[35] Kawsar, F., Nakajima, T., Park, J.H., Yeo, S.S.: Design and implementation of a framework for building distributed smart object systems. J. Supercomput. 54(1), 4–28 (2010).
[36] Mühlhäuser, M.:Smart products : an introduction .In: Constructing Ambient Intelligence: Am I 2007 Workshops, Springer, pp. 158–164 (2007)
[37] Fortino,G.,Guerrieri, A.,Lacopo,M.,Lucia, M.,Russo,W.:An agent-based middleware for cooperating smart objects. In: Corchado, J., Bajo, J., Kozlak, J., Pawlewski, P., Molina, J., Julian, V., Silveira, R., Unland, R., Giroux, S. (eds.) Highlights on Practical Applications of Agents and Multi-Agent Systems. Communications in Computer and Information Science, vol. 365, pp. 387–398. Springer, Berlin, (2013). http://dx.doi.org/10.1007/978-3-642-38061- 7_36
[38] Moumena, A. Chaqfeh and N.Mohamed, "Challenges in Middleware Solutions for the Internet of Things", in Conference: Collaboration Technologies and Systems (CTS), 2012 International Conference on 978-1-4673-1382-7/12/$31.00 ©2012 IEEE
[39] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V.Y. Terziyan, "Smart Semantic Middleware for the Internet of Things", in Proc. ICINCO-ICSO, 2008, pp.169-178.
[40] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim, "TinyREST: A protocol for integrating sensor networks into the internet," Proc. of REALWSN, Stockholm, Sweden, June 2005.
[41] A. Gómez-Goiri, D. López-de-Ipiña. "A Triple Space-Based Semantic Distributed Middleware for Internet of Things," In LNCS Vol. 6385, pp. 447-458. Springer, July 2010.
[42] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, V. 1736 Trifa, "SOA-based integration of the internet of things in enterprise services," in: Proceedings of IEEE ICWS 2009, Los Angeles, Ca, USA, July 2009.
[43] K. Aberer, M. Hauswirth, A. Salehi, "Middleware support for the Internet of Things," In: 5th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Berlin, Germany, Sep. 2006.
[15] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello, "Building the internet of things using RFID: the RFID ecosystem experience," IEEE Internet Computing, 13(3):48–55, 2009.
[44] Fosstrak: Open Source RFID Software Platform available at http://www.fosstrak.org/ accessed on April 28, 2011
[45] L. Roalter, M. Kranz, and A. M¨oller. "A middleware for intelligent environments and the internet of things," In Ubiquitous Intelligence and Computing, volume of LNCS, pp. 267–281. Springer Berlin / Heidelberg, 2010.
[46] P.-I. Blasco, F.-D. Rio, M.-C.RomeroTernero, D.Cagigas- Muñiz, and S.Vicente-Diaz. ''Robotics software frameworks for multi-agent robotic systems development''. Robot. Auton. Syst., 60(6):803–821, June 2012.
[47] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE: a software framework for developing multi-agent applications: lessons learned, Information & Software Technology 50 (1–2) (2008) 10–21.
[48] Bo Chen, Harry H. Cheng, Joe Palen, Mobile-c: a mobile agent platform for mobile c/c++ agents, Software: Practice and Experience 36 (15) (2006) 1711–1733.
[49] G. Metta, P. Fitzpatrick, L. Natale, Yarp: yet another robot platform, in: Software Development and Integration in Robotics, International Journal of Advanced Robotic Systems 3 (1) (2006) (special issue).
[50] Daniele Calisi, Andrea Censi, Luca Iocchi, Daniele Nardi, OpenRDK: a modular framework for robotic

software development, in: IROS, IEEE, 2008, pp. 1872–1877.

[51] J. L. de la Rosa, A.Oller, J.Vehi, & J.Puyol, Soccer team based on agent-oriented programming, Robotics and Autonomous Systems (21), p. 176, 1997.

[52] F. Mizoguchi, H. Ohwada, H. Nishiyama, & H. Hiraishi, Smart Office Robot Collaboration Based on Multi-agent Programming, Artificial Intelligence, Vol.114, No.1-2, pp. 57-94, 1999.

[53] J.Choi, Y.Cho,J. Choi, “Layered Middleware Architecture for Automated Robot Services”, in International Journal of Distributed Sensor Networks Volume 2014, Article ID 201063, 10 pages

[54] Young-Ho Suh, Kang-Woo Lee, Eun-Sun Cho, "A Device Abstraction Framework for the Robotic Mediator Collaborating with Smart Environments," 2013 IEEE 16th International Conference on Computational Science and Engineering, pp. 460-467, 2013 IEEE 16th International Conference on Computational Science and Engineering, 2013

[55] Anis Koubaa, Mohamed-Foued Sriti, Hachemi Bennaceur, Adel Ammar, Yasir Javed, Maram Alajlan, Nada Al-Elaiwi, Mohamed Tounsi, and Elhadi Shakshuki, “COROS: A Multi-Agent Software Architecture for Cooperative and Autonomous Service Robots” Cooperative Robots and Sensor Networks (2015 Edition) Springer, Studies in Computational Intelligence Series, 2015.

[56] Spiess, P.; Karnouskos, S.; Guinard, D.; Savio, D.; Baecker, O.; Souza, L.; Trifa, V. SOA-Based Integration of the Internet of Things in Enterprise Services. Proceedings of IEEE International Conference on Web Services (ICWS 2009), Los Angeles, CA, USA, 6–10 July 2009; pp. 968–975.

[57] C. Aguero and M. Veloso, “Transparent Multi-Robot Communication Exchange for Executing Robot Behaviors,” in Proc. of 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2012), ser. Advances in Intelligent and Soft Computing, vol. 156. Springer, April 2012, pp. 215–222

[58] L. Almeida, A. Santos, L. Oliveira, Structuring Communications for Mobile Cyber-Physical Systems, Chapter in, Management of Cyber Physical Objects in the Future Internet of Things: Methods, Architectures and Applications, Springer, ., 2016

[59] Borja, R., de la Pinta, J. R., Álvarez, A., Maestre, J.M. (2013), Integration of service robots in the smart home by means of UPnP: A surveillance robot case study, Robotics and Autonomous Systems, 61(2), pp. 153-160

[60] Guedes Lang, R., Da Silva, I.N., Romero, R., et al.: Development of distributed control architecture for multi-robot systems. In: 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), pp. 163–168. IEEE (2014)

[61] C.Hu, C.Hu, D.He, Q. Gu, “A New ROS-based Hybrid Architecture for Heterogeneous Multi-Robot Systems” in Published in 27th Chinese Control & Decision Conference (2015 CCDC); 2015, p4721-4726, 6p 2015

[62] Mouradian, C., Errounda, F., Belqasmi, F., & Glitho, R. (March de 2014). An infrastructure for robotic applications as cloud computing services. Internet of Things (WF-IoT), 2014 IEEE World Forum on, (pags. 377-382).