Towards a Cloud-assisted and Agent-oriented Architecture for the Internet of Things

Giancarlo Fortino and Wilma Russo
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)
Università della Calabria
Via P. Bucci, cubo 41C, 87036, Rende (CS), Italy
g.fortino@unical.it, w.russo@unical.it

***Abstract—* In the Internet of Things (IoT), all things (e.g. sensors, actuators, smart devices, smart objects, RFID, embedded computers, robots) have their identities, physical attributes, and interfaces. They will be seamlessly integrated into the information network such that they will become active participants in business, information and social processes wherever and whenever needed and proper. The technical realization of this vision is a complex challenge as distributed heterogeneous IoT components at different levels of abstractions need to cooperate among themselves, with conventional networked IT infrastructures, and also with human users. To cope with this issue, we propose the synergic exploitation of two complementary mainstream paradigms for large-scale distributed computing: the agent-oriented and the cloud computing paradigms. While the former can support the development of decentralized, dynamic, cooperating and open IoT systems in terms of multi-agent systems, the latter can empower the IoT objects with more computing and memory resources and effectively support system-wide higher-level mechanisms and policies. In this paper, we introduce a cloud-assisted and agent-oriented vision for IoT based on layered reference architecture. Finally, we briefly overview our agent-oriented middleware for cooperating smart objects and a sensor-cloud infrastructure that represent the basic building blocks for technically achieving such vision.**

***Keywords* - Internet of Things; Smart Objects; Agent-oriented Computing; Cloud Computing, Middleware**

I. INTRODUCTION

The Internet of Things (IoT) term usually refers to a world-wide network of interconnected heterogeneous objects (sensors, actuators, smart devices, smart objects, RFID, embedded computers, etc) uniquely addressable, based on standard communication protocols [1]. In the IoT, such objects have therefore their identities, physical attributes, and interfaces. They are seamlessly integrated into the information network such that they become active participants in business, information and social processes wherever and whenever needed and proper.

In particular, in this paper we model the IoT as a loosely coupled, decentralized system of cooperating smart objects (CSOs). A CSO is an autonomous, physical digital object augmented with sensing and/or actuating, processing, storing, and networking capabilities. CSOs are able to sense, store, and interpret information created within themselves and in the environment where they are situated, act on their own by also performing directed actuation, cooperate with each other, and exchange information with other kinds of IT devices/systems and human users.

The actual implementation and integration of IoT smart objects, their management as well as the development of real applications atop them, are complex challenges that require the synergic use of suitable paradigms and technology for large-scale distributed computing. To deal with this challenge, we propose the synergic exploitation of two complementary mainstream paradigms for large-scale distributed computing: (i) the *agent-oriented* paradigm, which fully supports the development of decentralized, dynamic, cooperating and open systems, and (ii) the *cloud computing* paradigm, which efficiently enables the empowering of computing and storage resources of IT systems.

The *Agent-oriented Computing* paradigm defines distributed software systems in terms multi-agent systems (MAS). In particular, agents are networked software programs that can perform specific tasks for a user and possess a degree of intelligence that permits them to perform parts of their tasks autonomously and to interact with their environment in a useful manner. The agent features perfectly fit the CSO features [2, 3]:

- *Autonomy*: agents/CSOs should be able to perform the majority of their problem solving tasks without the direct involvement of humans or other agents/CSOs, and they should have a degree of control over their own actions and their own internal state.
- *Interaction*: agents/CSOs should be able to interact, when they deem appropriate, with other software agents/SOs and humans to complete their own problem solving and support others with their activities where appropriate.
- *Responsiveness*: agents/CSOs should perceive their environment, in which they are situated and which may be the physical world, a user, a collection of agents, the Internet, and so forth, and respond in a timely fashion to changes that may occur in it.
- *Proactiveness*: agents/CSOs should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where and when appropriate.

The *Cloud Computing* paradigm provides flexible, robust and powerful storage and computing resources, which enables dynamic data integration and fusion from multiple data sources [4]. In addition a Cloud-based approach can offer flexibility and adaptability in the management and deployment of data analysis workflows. The dynamic deployment of software components as Cloud services removes the need for new client applications to be developed and deployed when the user requirements change. This also introduces an intrinsic competitive environment for the development of better services. Cloud computing layers (Infrastructure as a Service - IaaS, Platform as a Service - PaaS, Software as a Service - SaaS) and software components (e.g., databases, data mining, workflow tools) can be customized to support a distributed real-time system for the management and analysis of IoT objects and data streams generated by IoT objects.

This paper proposes a cloud-assisted and agent-oriented vision of IoT based on a layered reference architecture. Furthermore, we briefly overview our agent-oriented middleware for CSOs and BodyCloud, a sensor-cloud infrastructure, that represent the basic building blocks, which will be purposely integrated, for technically achieving such vision.

The rest of this paper is organized as follows. Section II provides a look at glance of the mainstream IoT visions. In Section III our reference architecture for cloud-assisted and agent-oriented vision of IoT is proposed. Section IV overviews our agent-oriented middleware for cooperating smart objects and a sensor-cloud infrastructure, and discusses their integration requirements to realize the proposed reference architecture. Finally conclusions are drawn.

II. SMART OBJECT-ORIENTED IOT

IoT semantically means a world-wide network of interconnected things (an ecosystem of things), which are uniquely addressable and based on standard communication protocols [1]. Things include sensors, actuators, sensor networks, embedded systems, RFID tags and readers, and other soft sensors in different forms. These things can be deployed in different physical environments to support diversified applications domains. They are communication-oriented objects and provide identification and information storage (e.g. RFID tags), information collection (e.g. sensor networks), information processing (e.g. embedded devices and sensor networks), and control and actuation (e.g. embedded systems including smart actuators). The interesting advantage is that everything is "reachable" and can be "exploited". The main disadvantages are that such enormous heterogeneity makes distributed communication and management very complex; moreover, "intelligence" is not embedded and should be provided at a higher level by means of smart services and/or applications.

Beyond such low-level and network-oriented vision of IoT, the smart object-oriented vision is at a higher level of abstraction and promotes an ecosystem of smart objects based on the Internet [5]. In particular, in such vision IoT is viewed as a loosely coupled, decentralized system of smart objects (SOs), which are autonomous physical/digital objects augmented with sensing/actuating, processing, and networking capabilities. Although, in such vision, not all "things" can be directly exploited as the object granularity is coarser, communication among smart objects is homogenized by the adoption of the Internet protocols and "intelligence" is mainly embedded inside the objects themselves.

Figure 1 shows a high-level layered architecture for the smart object-oriented IoT vision, where the main layers are: *Application*, *Middleware*, *Internet* and *Smart Object*. In particular:

- The *Application* layer encompasses applications based not only on SOs but also on other IT infrastructures.
- The *Middleware* layer provides as set of mechanisms for the naming, discovery, high-level interaction and state management of SOs.
- The *Internet* layer includes application, transport, and network protocols for supporting the communication with SOs and among SOs.
- The *Smart Object* layer offers programming frameworks and tools enabling the design and implementation of SOs.

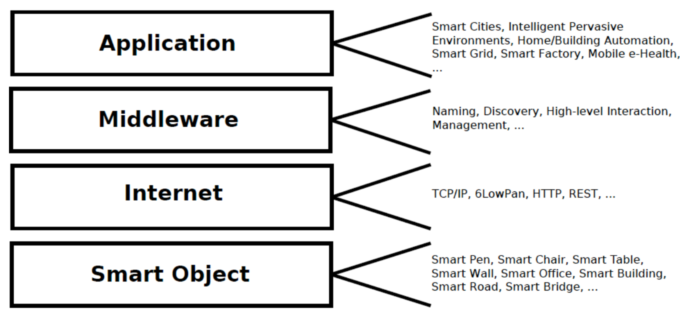

Figure 1. SO-oriented IoT architecture

Even though the standardization process of the SO communications based on IP is supported by the IP for Smart Objects (IPSO) alliance [6], the availability of middleware and frameworks that support development and management of SOs is still limited. Nowadays, apart from several available middlewares for smart environments (e.g. Smart-Its, 2WEAR, Ambient Agoras, Aura, Gaia, iRoom) [7, 8] that are not centered on the very concept of SO, a few SO-specific middlewares have been so far proposed: UbiComp [9], FeDNet [10], Smart Products [11], and ACOSO [12, 13].

UbiComp [9] defines a paradigm providing conceptual abstractions, the plug/synapse model and a middleware named GAS (Gadgetware Architectural Style)-OS, which is installed on each SO, to manage SO as components of distributed applications composed of ubiquitous computing services.

FeDNet [10] is based on a data-centric approach. Specifically, FeDNet uses XML-based documents to describe the requirements of an SO application, without considering the management of the SOs. Therefore, the services offered by SOs are described through structured documents. On the basis of such documents, the run-time FeDNet infrastructure provides a semantic association between the applications and the SOs. In FeDNet, SOs are objects of the daily life with computing and communication capabilities. However, as SOs are not proactive, task-oriented FeDNet applications are able to provide proactivity by SO orchestration.

Smart Products [11] aim at the development of SO equipped with proactive knowledge. Such knowledge is exploited to communicate and cooperate with human users, other smart objects and the external environment. SOs rely on MundoCore, a communication middleware, and a set of well-defined ontologies to enable effective cooperation among SOs.

ACOSO (Agent-based COoperating Smart Objects) [12, 13] is a middleware providing an agent-oriented programming model for CSOs and tools for their effective development. CSOs are based on an event-driven proactive architecture and on two different communication models (message passing and publish/subscribe). ACOSO is currently available atop JADE [14] and JADEX [15]. A more detailed description of ACOSO is reported in Sect. IV.A.

III. A CLOUD-ASSISTED AND AGENT-ORIENTED ARCHITECTURE FOR IOT

To deal with the complexity of the SO-based IoT, we propose a high-level architecture based on cloud and agent-based computing. The architecture named CA-IoT (Cloud-assisted and Agent-based IoT) is shown in Figure 2. The architecture components are:

- The *Smart User Agent*, which models human users in the context of specific smart systems. They therefore provide GUI-based functionalities through which users can request services and/or formalize specific service requests.
- The *Smart Interface Agent*, which defines an interfacing agent such as brokers, mediators, wrappers. Specifically, they are able to interact with and/or wrap components of the external IT systems.
- The *Smart Object Agent*, which models a CSO.
- The *CyberPhysical Environment*, which refers to the non-agent-oriented logical and physical context (made up of logical and physical components) in which agents are embedded. It can be modeled in terms of a reactive/proactive environment abstraction [16] that is able to interact with agents according to a specific coordination model.
- The *Cloud Computing Platform*, which supports all smart agents, empowering their specific resources, and allows for the definition of new (virtual) smart object agents as meta-aggregation of existing smart object agents.

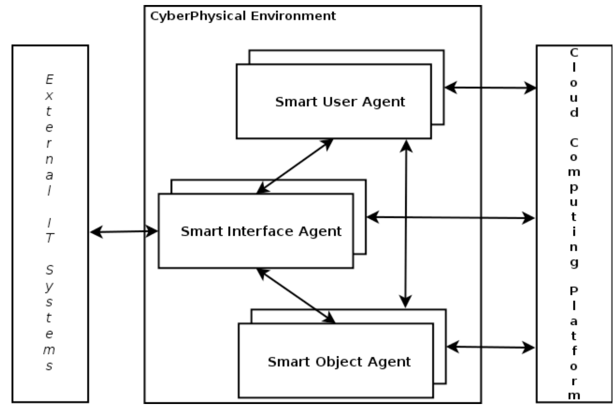

Figure 2. High-level architecture for Cloud-assisted and Agent-based IoT

IV. INTEGRATING SMART OBJECT MIDDLEWARE AND CLOUD PLATFORM FOR LARGE-SCALE IOT SYSTEMS MANAGEMENT

A. An Agent-oriented Middleware for the Development and Management of CSOs

The ACOSO middleware allows for the development and management of CSOs, which are modeled as agents that can cooperate with each other and with non-agent cyber-physical entities to fulfill specific goals. An ecosystem of CSOs therefore forms a multi-agent system (MAS). ACOSO currently relies on JADE that provides an effective agent management and communication support and on an external smart object discovery service that is purposely integrated into ACOSO [17]. Specifically, CSOs can be implemented as either JADE or JADEX agents, atop both Java-based and Android-based devices, and can cooperate by a direct coordination model based on ACL message passing and/or by a spatio-temporal decoupled coordination model relying on a topic-based publish/subscribe mechanism. Figure 3 shows the JADE-based ACOSO platform for CSO development and deployment. The platform is composed of three layers:

- The *high-level CSO architecture*, which is the reference architectural agent-oriented model for CSOs.
- The *JADE-based agent middleware*, which provides an implementation of the high-level CSO architecture through JADE and JADEX atop different computing devices (PCs and mobile devices).
- The *WSAN programming and management*, which is based on the Building Management Framework (BMF[1]) [18], for the management of the wireless sensor and actuator network (WSAN) of smart space-oriented CSOs, and on the Signal In-Node Processing Environment (SPINE[2]) [19, 20], for the management of the body area network (BAN) of body/personal oriented CSOs.

[1] http://bmf.deis.unical.it
[2] http://spine.deis.unical.it

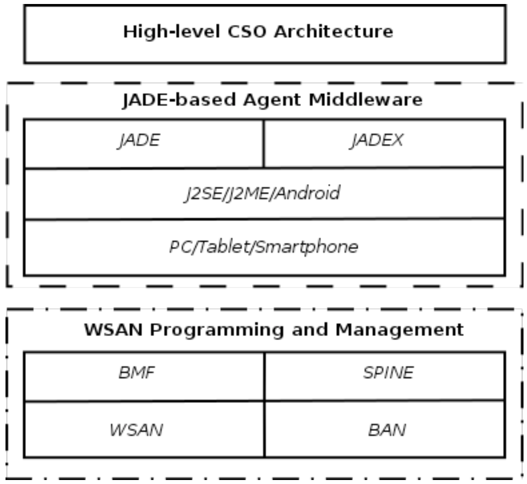

Figure 3. Agent-based Platform for Smart Object Development

Figure 4 shows the JADE-based CSO architecture. CSOs are agents of the JADE platform (either simple JADE agents or encapsulated JADEX agents). They are therefore managed by the AMS (Agent Management System), communicate through the ACL-based message transport system, and use the Directory Facilitator (DF) to look up CSOs and other agents. The DF, which supports dynamic agent service discovery, was purposely modified/extended in ACOSO to allow searching CSOs on the basis of their specific metadata: type, offered services, location, static hw properties, dynamic system properties. Currently, the extended JADE DF receives requests from the JADE-based CSOs and fulfills them by using the remote interface of an external CSO discovery service [17]. The main CSO architecture components are:

- The *Task Management Subsystem*, which manages the reactive and proactive tasks of CSOs. In particular, tasks are event-driven and state-based software components encapsulating specific objectives to fulfill through computation, communication, sensing/actuation, and storage management operations. Tasks can be defined as JADE Behaviours or JADEX Plans so their execution is based on the mechanisms provided by the basic JADE behavioral execution model or plan-oriented Jadex execution model, respectively. Being tasks driven by events, external CSO communication, signals to/from the CSO devices, data to/from the knowledge base (KB) are internally formalized and managed as events.
- The *Communication Management Subsystem*, which provides a common interface for CSO communications. In particular, message-based communication is based on the ACL-based MessagingService whereas publish/subscribe coordination is the TopicManagementService. The subsystem is internally organized in handlers. The CommunicationManagerMessageHandler, which is implemented as Behaviour in JADE and as Plan in Jadex, captures the ACL messages targeting CSOs and translates them into internal events. Moreover, the TCPAdapter and UDPAdapter manage communication with external networked entities based on TCP and UDP, respectively.
- The *Device Management Subsystem*, which manages the sensing/actuation devices that belong to the CSO. It is organized in a DeviceManager handling several DeviceAdapters. Currently, two DeviceAdapters are available: the BMFAdapter, which allows to manage WSANs based on BMF, and the SPINEAdapter, which allows to manage BANs based on SPINE. BMF and SPINE are based on IoT standards protocols such as IEEE 802.15.4, ZigBee, and 6LowPan.
- The *KB Management Subsystem*, which supports CSOs through a knowledge base (KB). It consists of a KBManager, which manages and coordinates different KBAdapters, and a KBAdapter, which manages a KB containing the knowledge of the CSO. KB can be local and/or remote and archives information that can be shared among tasks.

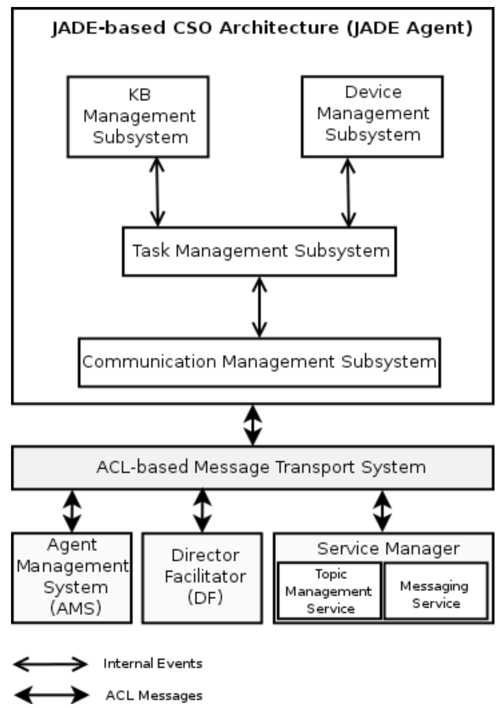

Figure 4. JADE-based CSO Architecture

Moreover, to support small-size CSOs running on a single sensor node, the Mobile Agent Platform for SunSPOT (MAPS[3]) framework [21] can be adopted to implement the CSO architecture. It is worth noting that MAPS agents can be wrapped by JADE agents through the JADE-MAPS gateway [22] so as to interact with JADE-based CSO agents.

B. *BodyCloud: an architecture for Cloud-assisted body area networks*

BodyCloud [23, 24, 25] is a SaaS-oriented architecture for the integration of BANs and a Cloud PaaS infrastructure. Its

[3] http://maps.deis.unical.it

architecture, shown in Figure 5, consists of four main subsystems (or sides):

- The *Body-side* manages the BAN and sends the collected sensor data to the Cloud-side through an Android-enabled mobile device. In particular, data acquisition is currently based on Android-SPINE, the Android version of SPINE. It allows Android-enabled smartphones and tablets to be used as coordinator of the BAN. In particular, the coordinator communicates with the wearable sensors by means of the SPINE application-level protocol atop Bluetooth and supports sensor discovery, sensor configuration, in-node processing, BSN activation/deactivation, data collection, and logging.
- The *Cloud-side* provides data collection and storage, processing/analysis and visualization. Each specific application can be defined through four programming abstractions: Group, Modality, Workflow/Node, and View. *Group* formalizes a specific application manipulating a well-defined BAN data source. *Modality* captures a specific interaction Body-Cloud sides and Viewer-Cloud sides. In particular, it models a specific service, such as data feeds to the Cloud-side, data analysis tasks, single-user or community applications. *Workflow* formalizes a data-flow process analyzing input data to generate output data. It consists of one or more nodes organized in a directed acyclic graph. *Node* is a specific algorithm that can be developed according to the Workflow Engine library (see Analyst-side in Fig. 5). A Node is uploaded to the Cloud-side where it can be used in different workflows. Finally, *View* defines the visualization layout of the output data for Viewers. Such abstractions are supported by the developed RESTLet-based SaaS Framework, which makes the interaction with the Cloud-side fully based on HTTP get, put, post, delete methods. The Cloud-side is supported by the Google App Engine PaaS[4] that enables data persistence and task execution.
- The *Analyst-side* supports the development of new application services. In particular, developers can create new BodyCloud services by defining the aforementioned abstractions. To program workflows, the Analyst-side is based on an appropriate development environment (XML Editor and Workflow Engine API).
- The *Viewer-side* visualizes the output produced by the data analysis through advanced graphical reports. By applying a View abstraction (see above) to the data, the graphical view is automatically generated. The current prototype is based on a Java library named jxReport that has been purposely implemented and integrated into the client application.

[4] https://cloud.google.com/products/

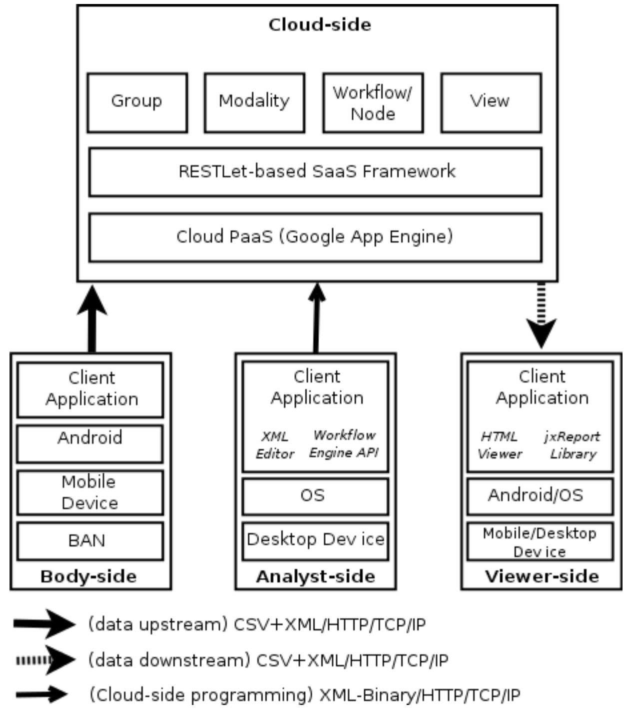

Figure 5. The architecture of BodyCloud

C. Integration requirements

To support the CA-IoT architecture introduced in Sect. III and thus develop large-scale IoT systems, ACOSO and BodyCloud are being jointly extended and integrated according to the following main requirements:

- *Smart agent enhancement*. While the basic smart agent layer is fully supported by ACOSO, the Cloud platform needs to provide a new functionality to define Cloud-based smart agents to dynamically create new virtual smart interface and object agents than run on the Cloud-side and seamlessly interact with the basic ACOSO agents.
- *Smart object data stream collection and management*. Data streams coming from highly decentralized smart objects needs to be efficiently uploaded onto the Cloud-side and here effectively managed.
- *Workflow-oriented analysis of smart object data*. Decision making applications should be dynamically developed through distributed workflows defined at the Cloud-side involving smart agents and cloud services.
- *Effective multi-level security architecture* for smart object data collection (from smart objects to the Cloud-side) and data analysis services (at Cloud-side).

V. Conclusions and Future Work

In this paper, we have proposed a novel high-level architecture for smart object-oriented IoT based on Cloud and Agents. Through the exploitation of agent-based computing, the proactiveness and cooperation capabilities of smart objects could be effectively defined, whereas by means of cloud computing, smart objects could be empowered in terms of processing power and storage resources. Thus, cloud-assisted and agent-oriented smart objects could be used as basis for the development of large-scale IoT applications and systems. The proposed architecture can be actually implemented by adopting and integrating an effective smart object middleware and an efficient cloud platform. To this purpose, we briefly introduced ACOSO, an agent-oriented middleware for the programming and management of cooperative smart objects, and BodyCloud, an architecture for the integration of sensors on the Cloud. They are currently being integrated according to specific defined requirements to achieve a large-scale distributed platform for smart object-based IoT development. Finally, future work will be aimed at defining a novel methodology for the development of IoT applications. Such methodology will be obtained as extension of ELDAMeth [26, 27], a methodology for simulation-based prototyping of distributed agent systems.

ACKNOWLEDGMENT

Authors wish to thank Antonio Guerrieri and Michele Lacopo for their valuable efforts and contribution in the development of the smart object agent-oriented middleware and Giuseppe Di Fatta (University of Reading) for his valuable insights for the BodyCloud approach definition. This work has been partially supported by TETRis – TETRA Innovative Open Source Services, funded by the Italian Government (PON 01-00451). Giancarlo Fortino is also partially funded by DICET – INMOTO – Organization of Cultural Heritage for Smart Tourism and REal – Time Accessibility (OR.C.HE.S.T.R.A.) funded by the Italian Government (PON04a2_D).

REFERENCES

[1] D. Bandyopadhyay and J. Sen, “The internet of things - applications and challenges in technology and Standardization,” Springer International Journal of Wireless Personal Communications, 58(1), pp. 49-69, May 2011.

[2] M. Vinyals, J. A. Rodriguez-Aguilar, J. Cerquides, “A Survey on Sensor Networks from a Multiagent Perspective,” The Computer Journal, 54(3), pp. 455-470, 2010.

[3] A. Rogers, D. Corkill, and N.R. Jennings, N. R. “Agent technologies for sensor networks,” IEEE Intelligent Systems, 24, pp. 13-17, 2009.

[4] R. Hill, L. Hirsch, P. Lake, S. Moshiri, "Guide to Cloud Computing. Principles and Practice," Computer Comm. and Networks, Springer, 2013.

[5] G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton, "Smart Objects as Building Blocks for the Internet of Things," IEEE Internet Computing, 14(1), pp. 44-51, January/February, 2010.

[6] Adam Dunkels and JP Vasseur, “IP for Smart Objects”, Internet Protocol for Smart Objects (IPSO) Alliance White paper #1, Sept. 2008.

[7] G. Fortino, A. Guerrieri, W. Russo, "Middleware for Smart Objects: State-of-the-art and Research Challenges", in *Internet of Things based on Smart Objects: technology, middleware and applications*, Springer Series on the Internet of Things. 2014. *to appear*.

[8] L. Roalter, M. Kranz, and A. Moller, A Middleware for Intelligent Environments and the Internet of Things," in Proceedings of the 7th international conference on Ubiquitous intelligence and computing (UIC). Berlin, Heidelberg: Springer-Verlag, 2010, pp. 267--281.

[9] Christos Goumopoulos and Achilles Kameas, “Smart Objects as Components of UbiComp Applications,” International Journal of Multimedia and Ubiquitous Engineering, Vol. 4, No. 3, July, 2009.

[10] F. Kawsar, T. Nakajima, J. H. Park, and S. S. Yeo, "Design and implementation of a framework for building distributed smart object systems," Journal of Supercomputing, vol. 54, no. 1, pp. 4-28, Oct. 2010.

[11] M. Miche, D. Schreiber, D., and M. Hartmann, “Core Services for Smart Products,” In: AmI-Blocks'09, at AmI'09, 2009.

[12] G. Fortino, A. Guerrieri, and W. Russo. Agent-oriented smart objects development.In Proc. of IEEE 16th Int. Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 907--912, 2012.

[13] G. Fortino, A. Guerrieri, M. Lacopo, M. Lucia, and W. Russo, "An Agent-based Middleware for Cooperating Smart Objects", in Highlights on Practical Applications of Agents and Multi-Agent Systems, Communications in Comp. and Inform. Science (CCIS), Vol. 365, pp. 387-398, Springer, 2013.

[14] F. Bellifemine, A. Poggi, and G. Rimassa, “Developing multi agent systems with a FIPA-compliant agent framework,”. Software Practice And Experience 31, 103-128, 2001.

[15] A. Pokahr, L. Braubach, W. Lamersdorf, “Jadex: A BDI Reasoning Engine,” In Multi-Agent Programming: Languages, Platforms and Applications. Multiagent Systems, Artificial Societies, and Simulated Organizations, Vol. 15, Springer, pp. 149-174, 2005.

[16] D. Weyns, A. Omicini, and J. Odell, Environment as a first-class abstraction in multiagent systems, International Journal on Autonomous Agents and Multi-Agent Systems 14 (1), pp. 5-30, 2007.

[17] G. Fortino, M. Lakovic, W. Russo, P. Trunfio, “A discovery service for smart objects over an agent-based middleware”, IDCS 2013, Lecture Notes in Computer Science (LNCS), Vol. 8823, pp. 1-14, 2013.

[18] G. Fortino, A. Guerrieri, G. O’Hare, A. Ruzzelli, “A Flexible Building Management Framework based on Wireless Sensor and Actuator Networks”, Journal of Network and Computer Applications, 35(6), pp. 1934-1952, 2012.

[19] F. Bellifemine, G. Fortino, R. Giannantonio, R. Gravina, A. Guerrieri, M. Sgroi, “SPINE: A domain-specific framework for rapid prototyping of WBSN applications” Software Practice and Experience, 41(3), 2011, pp. 237-265.

[20] F. Bellifemine, F. Aiello, G. Fortino, S. Galzarano, R. Gravina, “An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks”. Journal of Engineering Applications of Artificial Intelligence, Vol. 24, n. 7, pp. 1147-1161, 2011.

[21] F. Aiello, G. Fortino, R. Gravina and A. Guerrieri, A Java-based Agent Platform for Programming Wireless Sensor Networks, The Computer Journal, 54(3), pp. 439-454, 2011.

[22] M. Mesjasz, D. Cimadoro, S. Galzarano, M. Ganzha, Giancarlo Fortino, M. Paprzycki “Integrating JADE and MAPS for the development of Agent-oriented WSN-based distributed applications”, In Intelligent Distributed Computing VI, Springer, SCI series, Vol. 446, pp. 211-220, 2013.

[23] G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta, “BodyCloud: A SaaS Approach for Community Body Sensor Networks,” in *Future Generation Computer Systems*. 2014. to appear.

[24] G. Fortino, G. Di Fatta, “Engineering Large-Scale Body Area Networks Applications”, accepted 8th International Conference on Body Area Networks (BodyNets), Sept. 30–Oct- 2, 2013 Boston (USA), ACM press.

[25] G. Fortino, M. Pathan, G. Di Fatta, "BodyCloud: Integration of Cloud Computing and Body Sensor Networks", IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom 2012), Taipei, Taiwan, Dec 3-6, 2012.

[26] G. Fortino, A. Garro, S. Mascillaro, W. Russo, “Using Event-driven Lightweight DSC-based Agents for MAS Modeling,” in International Journal on Agent Oriented Software Engineering, 4(2), 2010.

[27] G. Fortino,W. Russo, “ELDAMeth: A Methodology For Simulation-based Prototyping of Distributed Agent Systems”, Information and Software Technology, 54(6), pp. 608-624, 2012.