Future Generation Computer Systems 102 (2020) 1038–1053

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Agent-based Internet of Things: State-of-the-art and research challenges

Claudio Savaglio [a,*], Maria Ganzha [b], Marcin Paprzycki [c], Costin Bădică [d], Mirjana Ivanović [e], Giancarlo Fortino [a]

[a] DIMES, Università della Calabria, Rende, Italy
[b] Warsaw University of Technology, Warsaw, Poland
[c] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
[d] CIT, University of Craiova, Craiova, Romania
[e] DMI, University of Novi Sad, Novi Sad, Serbia

ARTICLE INFO

Article history:
Received 12 May 2019
Received in revised form 9 August 2019
Accepted 9 September 2019
Available online 13 September 2019

Keywords:
Software agents
Internet of Things
IoT

ABSTRACT

The disruptive potentials of the Internet of Things (IoT) entails multifaceted requirements and development issues (large scale deployments, heterogeneity, cyberphysicality, interoperability, distributed smartness, self-management, etc.). To adequately tackle them and to comprehensively support the development of the IoT ecosystem, the Agent-Based Computing (ABC) represents a proper and solid modeling, programming and simulation paradigm. Indeed, abstractions, design methods, technology and frameworks related to the ABC have been widely exploited, possibly jointly with other well-established/emerging computing paradigms, to actually develop advanced IoT ecosystem. This survey, an extension of our previous work, reports most relevant contemporary contributions in the field, aiming at assessing suitability of the ABC paradigm for the (current and future) IoT development.

1. Introduction

In the last two decades, advancements in embedded processing, sensing, actuation, and wireless communication rapidly fueled the spread of novel cyberphysical artifacts and applications, among others, for the ambient assisted living and wellness, entertainment, logistic optimization, energy management, industrial automation. Indeed, simple movement detectors, radio-frequency tags, temperature sensors, but also more sophisticated smart gadgets, smartphones and smart vehicles, allow sensing the physical world, processing data, engaging customized users interactions and triggering actions over the surrounding environment. Also known as "Smart Objects" (SOs) [1] because of the (different degrees of) intelligence they exhibit, and being massively networked both on local and global scales, such heterogeneous devices communicate and cooperate with each other and with conventional systems, thus constituting the Internet of Things (IoT) [2] ecosystem.

The complex development of heterogeneous IoT ecosystems requires comprehensive support from different mainstream paradigms and approaches, especially from the closely related fields of wireless sensor networks, distributed systems, artificial intelligence, ubiquitous and pervasive computing [3]. In particular, the Agent-based Computing (ABC) [4] has been acknowledged as a comprehensive, effective enabler for cooperating, decentralized, dynamic and open IoT ecosystems. In this paper, we discuss how the ABC is, currently and effectively, exploited to develop IoT ecosystems. As matter of fact, the ABC provides ideas, metaphors, techniques, methods and tools for systematically conceptualizing, programming and simulating distributed systems composed of heterogeneous interacting entities [5]. Following an extended period of intense research and development, in the last decade the ABC seemed sidelined. Unexpectedly, the ABC recently found new interest and application in the IoT scenario and the marked interest of the research community in the agent-IoT duo [6] is demonstrated by the relevant number of publications materializing along this line (almost 1000 in the last three years, according to Scopus, just considering international peer-reviewed journals and conference papers; a detailed bibliometry is reported in Appendix) and the presence of the ABC among the IoT enablers outlined within the European H2020 Work Programme 2018-2020[1] as well as within some U.S.

---

* Corresponding author.
E-mail addresses: csavaglio@dimes.unical.it (C. Savaglio), m.ganzha@mini.pw.edu.pl (M. Ganzha), marcin.paprzycki@ibspan.waw.pl (M. Paprzycki), cbadica@software.ucv.ro (C. Bădică), mira@dmi.uns.ac.rs (M. Ivanović), g.fortino@unical.it (G. Fortino).

[1] https://ec.europa.eu/programmes/horizon2020/en/h2020-section/advanced-computing.

https://doi.org/10.1016/j.future.2019.09.016

National Science Foundations ongoing Programs[2,3] and awarded projects.[4,5] Such continuous research interest motivated us to extend our previous publication [7], from 2017, by surveying *(i)* the most up-to-date and relevant related work, proposing agents as IoT development enablers; *(ii)* the rise of synergies between the ABC and other mainstream paradigms/technologies, which are independent from IoT but beneficial for its development, e.g., Cloud and Edge Computing, Wireless Sensor Network, Machine Learning, Blockchain and Semantic technologies; and *(iii)* existing IoT applications and derivations, successfully enabled by the ABC, in novel fields like the Internet of Vehicles, Industrial IoT and Social IoT. Due to the extent of existing state-of-the-art material, providing an exhaustive analysis of all agent-based IoT contributions and/or agent-related IoT application scenarios seems unfeasible. Instead, we aim at highlighting main agents' pros and cons for the development of both SOs and IoT systems, surveying most relevant works.

The outline of this paper, graphically reported in Fig. 1, is as follows. Section 2, provides selected insights about the principal IoT development challenges and ABC distinctive features. Section 3 follows with a survey of several contributions exploiting the ABC to model, program and simulate SOs and IoT systems, including agent-based IoT methodologies. Next, in Section 4, selected synergies between the ABC and other technologies/paradigms (both emerging and well-established, but still instrumental for the IoT development) are presented, while in Section 5 examples of successful exploitation of agents in IoT applications are shown. Finally, the surveyed state-of-the-art is briefly analyzed, the open challenges discussed and some concluding remarks provided.

2. Background

2.1. Internet of things development issues and requirements

Multifaceted development issues must be faced, and related requirements addressed, in order to actually unfold the disruptive IoT potential [3]. *Heterogeneity* is the clearest issue featuring the kaleidoscopic IoT ecosystem [8], constituted by a plethora of different components (e.g., NFC tags, sensors, microcomputers, wearable gadgets, industrial robots, domotic devices, vehicles) and stakeholders (individuals, companies, public administrations). Therefore, providing *interoperability* to these heterogeneous subjects, connected through various communication technologies (Bluetooth, Zigbee, 3/4/5G, Wi-Fi, etc.), and deployed in networks of different scales (ranging from small-scale personal area networks up to large-scale industrial networks and extreme-scale, very dense metropolitan area networks), is a requirement necessary but challenging to meet. This is so, especially considering that the lack of established standards pushes towards poorly interoperable "intra-nets of things" [9]. Likewise heterogeneity, also *cyberphysicality* is entailed within the IoT concept. Indeed, the IoT is located at the border between virtual and real worlds, where SOs give rise to new cyberphysical functionalities and development issues, unforeseen or superficially treated by conventional computer engineering (cyber-physical security, personal data privacy, etc.). Hence, heterogeneity, interoperability and cyberphysicality motivate the use of multidisciplinary [10] and comprehensive *methodologies* [11]. These are used also for lowering time-to-market, efforts and probability of failure. The *size* and *rapid evolution* of IoT ecosystem are other two critical development issues given that, according to,[6] 20.4 billion of "things" will be networked in 2020. Operations, like things/services naming, discovery and deployment, would be definitively challenging in such a dense and dynamic scenario [2]. Therefore, *self-steering* and *decentralization* become imperative requirements for SOs and IoT systems, since their human-based, or centralized, management will bring result that are definitively unfeasible [12] (e.g., for the sake of bandwidth saving, an SO should be able, by design, to dynamically adjust its communication patterns; moreover, newly introduced SOs need to be automatically interfaced to and integrated into already deployed applications). Besides self-steering, SOs (which are different from simple resources such as RFID/NFC tags, databases, sensors and actuators) are expected to be *high-performing*, in terms of intelligence, reliability and context-awareness. Similar desiderata should be satisfied by all IoT systems that should expose autonomic, scalability and openness features [13]. Fulfilling these requirements would allow demanding IoT applications (e.g., augmented reality, industrial applications, emergency management, real-time systems) to provide high Quality of Experience (QoE) even in the presence of issues such as *SOs resource scarcity* or *poor infrastructures*. Indeed, without a satisfactory *usability*, the acceptance of novel SOs and IoT applications would be definitively compromised.

Summarizing, a number of development issues and related requirements makes the development of IoT ecosystems extremely challenging. Also other kinds of issues, concerning ethics, business and social sciences, are well known but disregarded because out of the scope of this paper. Nevertheless, the great potentials of IoT still make most of consumers, governments and companies seeing opportunities instead of threats.

2.2. Agent-based computing paradigm

Agent [4], as a sophisticated software abstraction defining an autonomous, social, reactive and proactive computational entity, is the key concept of the Agent-based Computing (ABC) paradigm. Likewise, MASs (Multi Agent Systems) [4] are ensembles of agents, which interact and cooperate in a certain environment (namely, the world of perceived resources), thus constituting distributed and self-steering societies featured by a strong situatedness and well-defined organizational relationships. Without claiming to be exhaustive, the above characterization already suggests that agent-related key abstractions allow for straightforwardly *modeling* complex systems, their components, interactions and organizational relationships, covering variety of domains (logistics, sociology, economy etc.).

Besides modeling, ABC provides also a well-established *programming* paradigm for implementing agents' advanced features, and effectively addressing key requirements, typical of modern distributed applications. Indeed, agent's, society's and environment's modeling abstractions have outlined a high-level, distributed programming paradigm, based on two milestones [14]: (1) encapsulation of control (each agent has its own reasoning capabilities and thread of control to expose context-aware and autonomous behaviors), and (2) interaction (coordination and cooperation typically based on high-level asynchronous message passing mechanisms). Indeed, shared communication standards and management specifications (like the Foundation for Intelligent Physical Agents FIPA [15] and Agent Communication Language ACL [16]) render agents interoperability facilitators and allow incorporating different resources and existing legacy systems within the agent society. By fostering computational efficiency, reliability, responsiveness, interoperability and scalability (particularly compared to centralized approaches), the agent-based programming paradigm allows implementing advanced applications while enhancing systems' performance.

Finally, agent-based systems can be straightforwardly simulated in order to study both emergent, individual patterns or

---

[2] https://www.nsf.gov/pubs/2018/nsf18557/nsf18557.html.

[3] https://www.nsf.gov/pubs/2019/nsf19566/nsf19566.html.

[4] https://govtribe.com/award/federal-grant-award/project-grant-1703782.

[5] https://www.nsf.gov/awardsearch/showAward?AWD-ID=1659774.

[6] https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

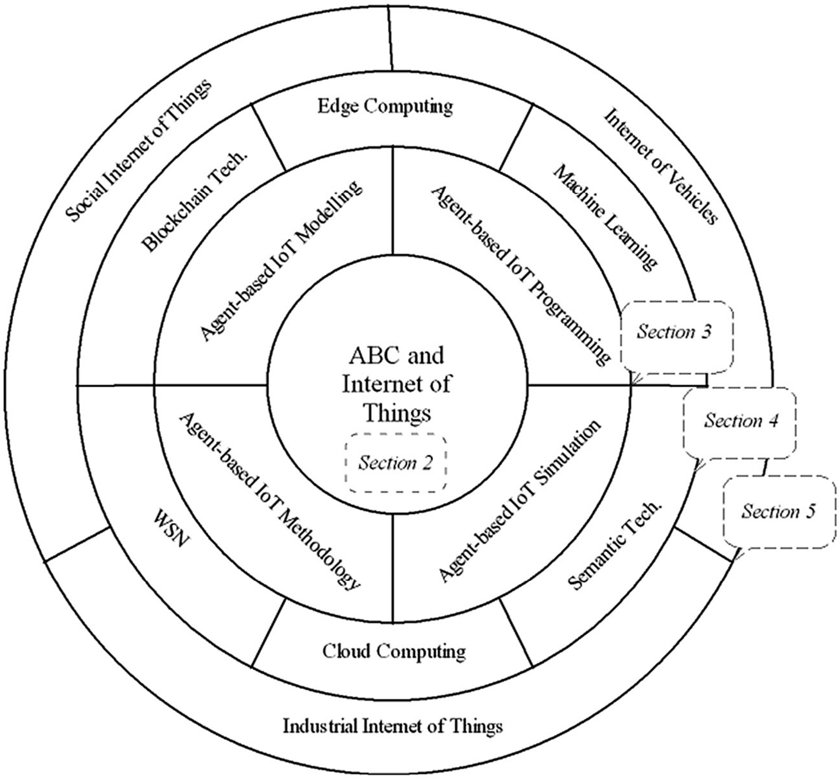


Fig. 1. Graphical outline of the survey.

collective phenomena [17]. By focusing on individual agents, their behaviors, and interactions (which reflect the ones of the real world), agent-based *simulation* is a natural approach for understanding and managing the global dynamic of complex systems, like distribution and supply networks, social sciences, etc. Indeed, even in highly scaling-up and interacting contexts, agent-based simulation facilitates the evaluation of distributed systems which expose discrete, not linear, adaptive behaviors.

Summarizing, the ABC is a rich and complex source of metaphors, techniques and tools: therefore, to provide a systematical approach to the agent-based modeling, programming, and simulation, several agent-oriented development *methodologies* have been designed and successfully applied [18]. However, as underlined in [19–21] and in Section 6, the ABC is neither a universal nor necessarily effective development solution, and its adoption needs to be carefully assessed. Indeed, both agent-level and society-level pitfalls can occur from different perspectives (e.g., management, conceptual, design) and these can outweigh every agent-related benefits.

3. Agents' contribution in development of IoT systems

Complex, dynamic, situated and autonomous systems, both natural and artificial, can be naturally approached by adopting the agent-oriented perspective [5]. Based on the strong conceptual alignment between agents/MAS and SOs/IoT systems [22], and according to the match between IoT development requirements and agent-related benefits, the ABC has been exploited to systematically drive and speed-up the development of SOs and IoT systems. An overview about the high-level modeling, programming, simulation and methodology targets of the agent-based IoT is shown in Fig. 2, while main contributions which exploit the ABC for modeling, programming and simulating purposes have been surveyed and compared in Table 1. This table summarizes findings contained in Sections 3.1–3.4, and indicates for each contribution if *(i)* a fine/coarse grained agent-based modeling of IoT entities is performed, *(ii)* mechanisms for (technological/ syntactical/ semantic) interoperability, autonomicity, cognitivity, virtualization or security are implemented, *(iii)* pure or hybrid activity simulation is supported, and *(iv)* an agent-based IoT development methodology is provided. The definitions of these *keywords* is reported in what follows.

3.1. ABC as IoT modeling paradigm

Key features of both SOs and IoT systems can be captured, at different degrees of granularity and in a technology-agnostic way, through the agent-based modeling. Indeed, the agent model naturally embeds SO autonomicity, proactiveness and situatedness, while other important SO features can be explicitly described through agent-related concepts. For example, in [23,49,56], SO functionalities are expressed as goals, SO working plans as behaviors, and SO augmentation devices (i.e., knowledge bases, sensors and actuators) as dynamically bindable agent resources. However, these working plans adopt different mechanisms for specifically characterizing SOs/agents. In [48,55], for example, SO/agent behaviors, goals and communications paradigms depend on their own role, which is taken from context-dependent repositories (e.g., for the transportation, smart car, smart driver-support, smart road). Likewise, in [24,25], each SO/agent complies to a template, which encodes plans and goals according to its functionality. Differently, there are some contributions, which do not provide a-priori defined roles or templates. In [23], each SO/agent has its own model (formalized as an automaton) which, on the basis of perceived stimuli (modeled as messages incoming from the environment or other entities), dictates the actions to be performed. Likewise, in [29,40], SOs/agents' behaviors encode both design goals (encapsulated in state-based tasks) and (re)actions to the risen internal and external events. Finally, real-time sensor data, position and resources availability

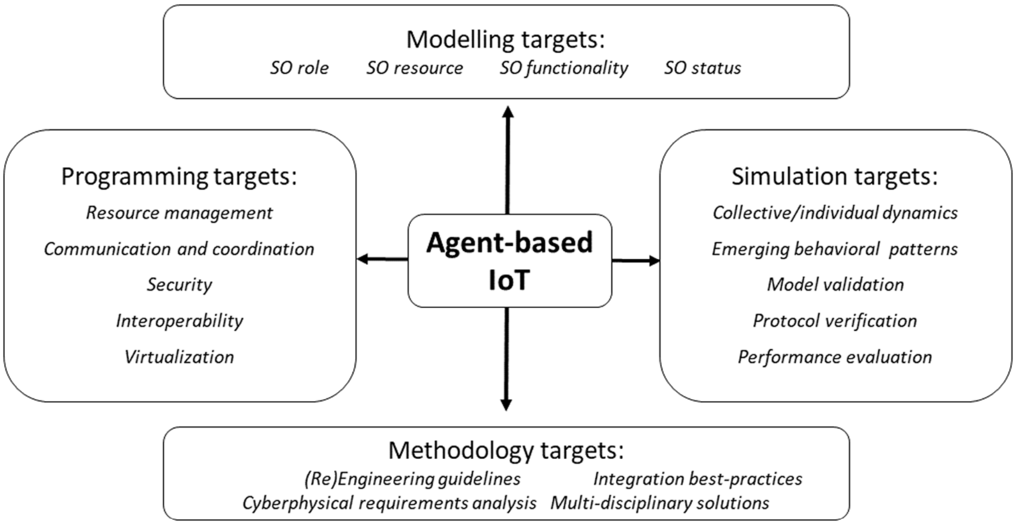


Fig. 2. High-level modeling, programming, simulation and methodology targets of the agent-based IoT.

Table 1
Relevant agent-based works with their features - *T*echnological, *Sy*ntactical, *Se*mantic interoperability; *A*utonomicity; *C*ognitivity; *V*irtualization; *S*ecurity.

Surveyed work ⟨name, ref.⟩	Agent-based IoT model		Agent-based IoT implementation							Agent-based IoT simulation		Agent-based IoT methodology
	Fine grained	*Coarse grained*	*T*	*Sy*	*Se*	*C*	*A*	*V*	*S*	*Pure*	*Hybrid*	
Cascadas, [23]	✓		✓	✓			✓					
iCore, [24]	✓		✓	✓	✓	✓			✓			
FIoT, [25]	✓		✓	✓		✓	✓	✓				
CACB, [26]		✓					✓	✓			✓	
[27]		✓					✓	✓			✓	
ACOSO [28], [11], [29], [13], [30]	✓		✓	✓	✓	✓	✓	✓			✓	✓
UBIWARE, [31]; UBIROAD, [32]	✓		✓	✓	✓	✓		✓	✓			
[33]		✓	✓	✓				✓				
[34]	✓		✓	✓	✓		✓	✓				✓
AoT, [35]		✓	✓	✓		✓	✓	✓				
Smart Grids, [36]		✓		✓						✓		
[37]		✓	✓	✓				✓			✓	
TAEC, [38]		✓	✓	✓	✓		✓	✓	✓			
CIoT, [39]	✓		✓	✓	✓	✓		✓	✓			
iSapiens, [40]	✓		✓	✓				✓	✓			
[41]		✓	✓	✓				✓				
Radigost, [42]		✓	✓	✓				✓				
ASSIST, [43]	✓		✓	✓	✓	✓		✓		✓		
BEMOSS, [44]		✓	✓	✓				✓	✓			
INTER-IoT, [45]	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓
VICINITY, [46]	✓		✓	✓	✓	✓		✓	✓			
SOL, [47]	✓		✓	✓	✓		✓					
[48]	✓		✓	✓	✓							
[49]		✓	✓	✓			✓					
[50]		✓	✓	✓		✓	✓					
Smart Santander, [51]		✓	✓	✓	✓			✓	✓			
[52]		✓	✓	✓	✓			✓	✓			
I-Room [53],		✓	✓	✓	✓		✓	✓				
Prometheus, [54]	✓		✓	✓	✓			✓	✓			✓
ASEME, [55]	✓		✓	✓	✓			✓	✓			✓
SAMSON, [56]	✓						✓			✓		
SenseSim, [57]	✓		✓	✓				✓			✓	
[58]		✓			✓	✓	✓			✓		
[59], [60], [61]		✓	✓	✓			✓	✓	✓	✓		
FABIoT, [62]	✓		✓	✓				✓			✓	

dynamically determine the state of the SO/agent in [49]. Apart from the specific modeling approach, all the surveyed agent-oriented approaches provide high-level models to effectively abstract principal SOs features transparently from a specific technology, thus supporting the preliminary development phase of analysis. However, besides the satisfactory "per-se" agent-based SOs modeling, further research efforts are necessary to thoughtfully model relationships among agents/SOs interacting within physical environments. For example, [63] describes opportunities and limitations (along with the current and potential socio-cultural impact) of a truly networked and cooperative SO-based IoT based on the "agency" metaphor.

Although concepts such as negotiation, competition, cooperation and delegation between agents and other entities, have complex definitions, the ABC still represents an effective IoT modeling paradigm, as well as a profitable baseline for the subsequent phases of agent-oriented IoT programming and simulations [11].

3.2. ABC as IoT programming paradigm

The marked heterogeneity of resources and communication protocols, featured in the IoT ecosystems, motivated an agent-oriented approach for programming uniform interfaces and transparently interacting with different SOs and resources. Software adapters, internally coordinated by a device manager [29] and purposely developed for target technologies, allow accessing and managing augmentation devices of SO/agent in [24–27,29,31,32]. Being developed as pluggable and customizable software components, these adapters ensure modularity and extensibility of system programming. A completely different approach is adopted in [44,48], in which specific agents are deputed to interface each resource with the related SO and with the system. Although transparent to technological heterogeneity, this solution does not work for those constrained devices unable to support a complete agent-based architecture.

Beyond facilitating resource management, agent-oriented programming promotes communication and coordination in the IoT ecosystem (i) directly, by implementing the IEEE FIPA 'de facto' standard specifications [15], and (ii) indirectly, by supporting the SOs *Virtualization* [24] for the sake of a major accessibility and integration of the agentified SOs. With respect to the first point, FIPA specifications provide standardized message format and content, an effective message transport service, and both semi/centralized and distributed services of agent discovery. The ACL [16] allows encoding the message envelope; the message content, instead, is typically expressed through metadata-oriented languages and ontology to facilitate both context management and data exchange. With respect to the second point, the functionalities of an agentified SO can be transparently accessed over standard, platform- and language-independent Internet protocols (e.g., SOA and REST [41,42]) and then exploited as a monolithic Web Service [35] or as an ensemble of small, loosely-coupled and distributed microservices [64]. Just microservices are currently on the hype due to the agility they provide for the development of a IoT system, and the agent-based programming can provide them further features like autonomy, social ability and elasticity [59–61].

In brief, agent-based programming fosters:

- *Technical interoperability*, by means of shared resources and communication interfaces [23,29,42,47,50,51] (though, in some cases, agents belonging to different organizations are not totally interoperable [53]);
- *Syntactical interoperability*, by means of a shared message format, because ACL is adopted across FIPA standard obeying platforms for message envelope, while XML and JSON are used for message content in [31,33,41,46,48,51] (but it is worth noting that ACL is a 'de facto' standard but not the only language [19]); and
- *Semantic interoperability*, by means of shared ontology and knowledge representation [24,31,32,39,52,65] (but this feature is quite limited and underdeveloped because of the shortage of grounded domain-specific ontology and semantics [20]).

At a higher level of conceptualization, agents allow to straightforwardly instill smartness and autonomy within a single SO, and realize cognitive and autonomic IoT systems [12,28]. Indeed, agent-based programming fosters (i) *Autonomicity*, by self-configuring, self-healing, self-protecting and self-optimizing both SOs and IoT systems which require minimum human interventions for their management [23,38,47,50]; and (ii) *Cognitivity*, in terms of context-aware and adaptive SOs and IoT systems [24,39,46], able to solve complex problems autonomously, if properly trained. Apart from ensuring self-management and distributed intelligence, autonomic and cognitive features are also key to implement advanced *Security*, for example through (un)conventional trust mechanisms (see Social IoT in Section 5) aiming at a secured large-scale IoT ecosystem.

3.3. ABC as IoT simulation paradigm

A number of contingent factors [56] (SOs density, network design, irregular traffic, wireless coverage, etc.), the cost of the hardware and its installation, along with the scarcity of professional installers, make the IoT ecosystem deployment complex, costly, error prone and time consuming, especially in large-scale scenarios (e.g., a smart city in which a coordinated, city-wide smart parking system could require very high budgets). Therefore, the simulation activity plays a crucial role [30], enabling the comprehension of collective/individual dynamics and the estimation, validation and verification of performance, models and protocol before the deployment phase.

Pure agent-based simulators allow effectively inspecting high-level aspects such as the rise of collective dynamics and behavioral patterns in large scale, distributed, event-based scenario [62,66]. JADE has been used in [67] to simulate a Smart City in which heterogeneous "agentified" SOs, able to act autonomously and collaborate, dynamically consume and/or produce energy. In [68], authors integrate agent-based simulation and evolutionary game theory to analyze cooperative patterns, dynamic processes and macro emerging actions in the IoT scenario. Through a set of simulations carried out on Anylogic, authors evaluate conditions leading certain business models to dominate the IoT market. In [69], the agent-based simulation focus is on IoT services, particularly to evaluate several discovery strategies considering different communication topologies. In [70], instead, authors propose an Agent-Oriented Petri Net to analyze the dynamic behavior of services by performing model checking and, hence, exhaustively and automatically check specifications and properties. Finally, Netlogo has been exploited in [58,71] to aggregate IoT services under complex users' requests and to manage real-time traffic information.

Although pure agent-based simulators can successfully validate SOs interactions and operations, they typically outline quasi aseptic simulation environments, distant from the real cyber-physical IoT ecosystem, because low-level communication and mobility issues are often neglected or coarsely handled [36,43]. Therefore, a novel research line foresees a hybrid approach, based on the joint exploitation of agent-oriented modeling and network-based simulation. In particular, agent's logic is implemented within event-based network simulators like OMNeT++ [22,28,30,37], ASSIST [43] and SenseSim [57], because SOs/agents interactions are asynchronously event-driven and time-dependent. Such hybrid simulation approach allows mitigating limitations of pure agent-based simulation (but maintaining advantages derived from ABC) and effectively simulating IoT systems of different scales (ranging from narrow local networks up to widespread Smart City), with different mobility patterns, interaction protocols, communication parameters, etc. [22,28,62,72]. In this way, both qualitative and qualitative aspects can be assessed, providing a preliminary and reliable overview of operation, of an IoT system to be developed.

Summarizing, few computing paradigms deal with IoT simulation (e.g., Aggregate Computing [73], SOA [74]) and the hybrid agent-based approach represents one of the more valid candidates. However, as opposed to the well-established modeling and programming, agent-based IoT simulation is at an early stage, while IoT-specific simulators are not currently available. Nevertheless, overlooking this aspect could represent a crucial pitfall compromising the agents acceptance in the IoT scenario [75], considering that, without a central control, unexpected and emergent behaviors are likely to appear within MASs [76].

3.4. Agent-based methodology for IoT

The peculiar features of the IoT context demand agent-oriented methodologies to be specifically extended [54,55], or ex-novo designed [11,34]. Indeed, conventional agent-based methodologies contemplate the usage of agent-based models, programming techniques and simulation tools but, at the same time, they are not straightforwardly applicable in the IoT scenario. Therefore, for comprehensively driving the IoT ecosystem development, the (few available) agent-based methodologies for IoT:

- thoroughly face the cyberphysicality of SOs, providing by design solutions for interoperability, security and scalability [11,34,54];
- seamlessly glue hardware and software components by defining, typically at the middleware layer [77,78], suitable coordination, virtualization and management techniques [11];
- jointly analyze specific SOs and IoT system requirements [34,55] with the infrastructural features and limitations of the development context;
- accommodate different perspectives and expertise, exploiting both technical (e.g., Unified Modeling Language [55] and Business Process Model Notation [54]) and not-technical (textual descriptions [34]) notations to depict the IoT ecosystem, its relevant use cases, users, services, stakeholders, etc.;
- drive and promote the integration of different computing paradigms for IoT (see Section 4) in different application contexts (see Section 5) through guidelines and best-practices, possibly unbound from specific protocols or technologies [11,45].

Without extensively facing such aspects, any development methodology, even if effective in conventional agent-based context (for example, Tropos [79]), would fail in supporting the IoT ecosystem development. Finally, it is worth noting that agent-based methodology facilitates and speeds-up not only the development of novel SOs/IoT systems, but also the re-engineering of existing ones. Indeed, as shown in [80], agent-based methodologies can drive SO/IoT system re-engineering, in order to highlight and enhance both functional and non-functional features (e.g., support to interoperability, attention for resource-constrained SOs, modularity, maintainability, evolvability) which generate fundamental benefits for the complex, heterogeneous and constantly evolving IoT scenario.

4. Integration of agents with other paradigms and technologies for IoT

As reported in the previous Section, the ABC can enable the SOs' and IoT systems' development; moreover, the ABC is also prone to be beneficially integrated with other computing paradigms and technologies which play an important role within the IoT scenario. Previous, or contemporary to the IoT, these paradigms/technologies have their own independence, but they can effectively support the IoT development, also jointly with the ABC. Again, due to the extent of the research field, an exhaustive survey is not feasible. Therefore, we present some relevant mainstream paradigms and technologies which perfectly work in conjunction with the ABC to eventually support the IoT development. For each paradigm/technology, a brief introduction is provided, the contact points with IoT elicited and, finally, the benefits of synergies with the ABC discussed. Main findings of this analysis are summarized in Table 2.

4.1. Cloud computing

Cloud Computing empowers systems' resources and functionality by supporting extreme-scale intensive computation and massive, dynamic, heterogeneous data integration, storage and analytics. In the IoT context, Cloud Computing represents a fundamental enabler for the development of agent-based SOs, even if resource constrained. Indeed, Cloud Computing allows "agentified" SOs to locally provide complex functionality (enabled by SOs virtual aggregation and SOs services composition) while, behind the scenes, computation and data are offloaded on powerful remote servers. In such a way, SO hardware/software limitations are effectively and transparently mitigated.

In a Cloud-assisted and Agent-based IoT (CA-IoT) platform [81,82] wearable SOs, monitoring human activities, are first locally agentified and then virtualized into the Cloud, where incoming data is stored and analyzed, and new, empowered, virtual SOs created, as a meta-aggregation of existing ones. In particular, at the Cloud side, distributed data-flow processes analyze input data and dynamically support the development of new application services to be provided by the agents running on the SOs.

Similarly, authors of [83] present an agent-based traffic management system: the Cloud supports processing of complex traffic strategy and massive data transport by providing the needed storage and computing resources. In that way, highly demanding and interactive traffic simulations can be performed in a standard development environment (therefore, easily accessible by multiple agents/users like traffic managers, traffic participators, traffic-strategy developers, etc.) exploiting both real-time data collected by SOs and historical dataset. From another perspective, the ABC facilitates constructing software tools and testbeds to automate the management of both cloud's resources and services, to the advantage of single SO as well as IoT system performace. With respect to cloud's resources management and, particularly, authors of [84] propose an Autonomous Agent Based Load Balancing Algorithm (A2LB), for maximizing system's utilization, responsiveness, scalability, throughput and reliability. A set of agents is in charge of monitoring the load of every machine, controlling the transfer, selection and location policies, and proactively allocating the resources, aiming to remotely support SOs of the Cloud-assisted IoT system. With respect to cloud's services management, instead, in [85] authors designed and developed Cloudle, an agent-based search engine for human users and, virtually, SOs, supporting cloud service discovery, negotiation, and composition. Cloudle foresees the exploitation of agents for consulting cloud ontologies and automatically determining the similarity/compatibility among cloud services specifications and consumers'/SOs' requirements.

Table 2
Summary of Section 4 findings.

Paradigm\Technology integrated with the ABC	High-Level integration goals	Examples
Cloud Computing	Virtualization of (agentified) SOs; (agentified) SO's data and computation offloading; remote management of (agentified) SOs' resources	SOs aggregation techniques; load balancing algorithms for the IoT system; IoT system data analytics
Edge Computing	Providing (agentified) SOs with context-aware and autonomous behaviors; providing a decentralized, fine-grained control of (agentified) SO's and network resources	Cognitive algorithms and smart data aggregation techniques for SOs; dynamic scheduling of SO's tasks; SO-specific privacy-preserving techniques
Wireless Sensor Network	Easing the prototyping of (agentified) sensor nodes' software; effectively handling (agentified) sensor nodes' low level functionality and services; opportunistically managing (agentified) sensor nodes' and network's resources	On-node signal processing algorithms; efficient sensor's radio management; code mobility
Machine Learning	Increasing the learning and decision-support capabilities of (agentified) SOs; supporting the (agentified) IoT system's collective intelligence; enabling the context-aware, real-time reconfiguration of (agentified) SOs	automatically-built SOs' models; time series analysis techniques; evidence-based decision support systems
Blockchain Technology	Securing the decentralization of (agentified) SO's data and access control; improving (agentified) SOs accountability and identity management; enabling a trustworthy, decentralized coordination within (agentified) IoT systems	SOs' policies management; collaborative anomaly detection; agentified Smart Contracts and automatic Blockchain's transaction management
Semantic Technology	Assisting (agentified) SOs in a deeper and broader understanding of their environment; fostering (agentified) SOs reasoning through shared knowledge basis; facilitating the communication among heterogeneous (agentified) SOs	Translators for heterogeneous SOs data models; ACL messages with semantic payloads; domain-specific ontologies and vocabularies

4.2. Edge computing

Edge Computing moves computation and storage resources close to where data has been generated, thus providing context-awareness, responsiveness, privacy, robustness and efficiency (in terms of both bandwidth and energy consumption) to IoT applications, like video analytics, personal healthcare, autonomous vehicles, etc. However, as a decentralized computing paradigm, Edge Computing demands SOs to be autonomous, interoperable and smart, as far as allowed by their (often limited) software and hardware resources. And that is where ABC comes in, enabling SOs exhibiting autonomous, context-aware, smart behaviors and fostering the collaborative execution of Edge Computing-based IoT applications.

Context-awareness and autonomy are design goals of FLEC (Flexible Edge Computing) architecture [86], which leverages a MAS to autonomously and dynamically determine the *(i)* allocation of processing load and tasks assignment to the Edge and to the Cloud, according to the properties of applications, contexts, servers' and SOs' resource situations, and *(ii)* provisioning of services, suitable for each user in real time, by reflecting detailed information like user's behaviors, intentions, and preferences collected by IoT devices. Aiming at interoperability, instead, the ROAgent framework, described in [87] and [88], jointly exploits agent-oriented, Edge Computing and resource-oriented paradigms to enable the interaction of resource-constrained, heterogeneous, SOs within the IoT ecosystem through standard Web technologies. Indeed, SOs' resources and operations are exposed as programming language- and platform-independent Web services, which can be browsed and searched over the Internet.

Likewise, [89] exploits the ABC for developing IoT Smart Environments which exhibit reactive, proactive, and cognitive behaviors through computational resources located either in the Cloud or at the network Edge. In [90], the same authors propose the CEIoT platform, a Cognitive-enabled and Edge-based IoT architecture, which leverages on the ABC for the realization of decentralized cognitive algorithms and the versatile development of smart data aggregations.

Finally, CRESCO [91] is an agent-based framework provided with an application description language for supporting real-time streaming applications at the newtork Edge. CRESCO allows managing multitude of geographically distributed services based on heterogeneous resources, whose scheduling, provision and performances can be managed through proper high-order modules, while the ABC provides a fine-grained control over the structure, communication, and security protocols of the distributed system.

4.3. Wireless sensor networks

Wireless Sensor Network (WSN) is an enabling technology for the IoT, allowing the data collection, gathering, processing and forwarding through (typically resource constrained) wireless sensor nodes scattered across the monitored area. Rapidly spreading in the first 2000s, WSNs found application in multitude of scenarios (ambient assisted living, structural health, e-health, etc.), which later have become of interest also for the IoT. In that sense, WSNs can be considered one of the forerunners of the IoT systems as well as one of its essential constituent.

Because of the scarce resources of both infrastructures (bandwidth is typically quite limited in wireless networks compared to wired ones) and WSNs' nodes (limited computation, storage and energy), agents' features of smartness, mobility and autonomy can be highly beneficial both to functional (by instilling smartness) and non-functional (energy and bandwidth saving) scopes.

With respect to the first point, lightweight agents can be deployed on sensor nodes to efficiently handle their low-level functionality (sensing, filtering, pre-processing, storage, communication) and basic services (timer handling, resource access, etc.). This is the case of [92], in which an agent-oriented framework for real-time human activity monitoring is proposed. In particular, here, lightweight agents operate according to an Event-Condition-Action (ECA) automaton and aim at computing the accelerometric sensory data, aggregating their features, and recognizing user movements and postures. Moreover, the agent-oriented design and programming allows effective and rapid prototyping of the sensor software, which shows also good performance in terms of recognition accuracy and nodes synchronization.

With respect to the second point, agents deployed within WSNs allow saving both network bandwidth and sensors energy. The distributed, intelligent, decision making process offered by lightweight agents is exploited in [93] to drive the opportunistic activation of the sensor nodes, with a subsequent energy saving. In [94], instead, an agent-based architecture implements a distributed cognitive radio resource management framework which dynamically manages shared radio resources to minimize interference, and hence the energy waste. With the goal of minimizing the overall communication costs, sensory data can be processed locally rather than directly moved towards a sink. Following such principle, in [95] and [96] agents locally transform and reduce a large amount of sensory data by eliminating data redundancy among sensors through a context-aware on-node data processing (filtering and clustering, mainly).

4.4. Machine learning

Machine learning (ML) is recently becoming one of the fastest growing areas of computer science, with extensive possibilities for wide range of applications. Essential concepts, algorithms, and theoretical frameworks in ML predominantly include supervised and unsupervised learning, statistical learning theory, probabilistic graphical models and approximate inference with great possibilities for developing new and more powerful approaches. In fact, ML can be seen as excellent alternative and upgrade for the development of algorithmic solutions, as compared to conventional engineering approaches. Usually, ML refers to automated detection of meaningful patterns in big datasets. Combination of wide variety of ML algorithms and their powerful processing of big datasets collected from IoT, SOs and Smart Environments can offer, and achieve, great benefits in variety of applications in different domains. Additionally, parallelization can significantly speed-up ML algorithms working with big datasets produced in IoT and Smart Environments. Moreover, time-series (TS) analysis, as a specific part of ML techniques, can be extremely useful in smart and IoT environments. TS analysis is used to describe changes of observed phenomena over time, such as the sensory information collected by SOs. Different systems and tools like R and FAP [97] can be used to enhance diverse aspects of smart environments and emergency scenarios [98], for example smart cities.

Agents, and their key abilities of learning, offer a significant shift in employing ML functionality in intelligent decision making processes. During the last two decades, developments in the agent technologies and ML have become complementary and agent-based platforms utilizing ML for intelligent decision support and automation have been constantly developing [99]. A set of agents, in smart environments, employing ML, may increase efficiency of learning and decision-support. Similarly, collective computational intelligence can support several ML algorithms, where a synergistic effect is expected from combining efforts of various kinds of agents. In essence, a set of agents, cooperating in distributed smart environments, can increase performance of such systems. For example CityMatrix [100] is an urban decision support system that facilitates evidence-based and more collaborative urban decision making. Role of applied ML techniques is to support real-time prediction of an agent-based urban traffic model. The system provides efficient optimization feedback, in real-time, to support the decision making process. Another interesting approach that incorporates agents, IoT and ML is presented in [101]. Authors propose ML to assists developing embodied and self-configurable agents for the IoT. In particular, a feedback-evaluative ML enables the reconfiguration of a MAS on the basis of the environmental context.

4.5. Blockchain technology

Blockchain technology is an emerging and promising approach for securing the decentralization of data and control, as well as for executing Smart Contracts (SCs), namely trusted and automatic activities which encapsulate arbitrary and stateful functionality. Ranging from healthcare to logistics, from ambient assisted living to energy-trading, a plethora of large-scale and privacy-sensitive distributed IoT systems (with their often resource-constrained SOs) can greatly benefit from Blockchain Technology when performing key operations like object tracking, identity and policies management, transaction traceability and accountability, coordination, etc. By making such operations more secure, autonomous, flexible and even profitable, fostering also scalability, cross-organizational collaboration, interoperability, Blockchain Technology can fulfill some of the most important requirements and features shared by both multi-agent and IoT systems such as data integrity, privacy, authenticity, big data management and decentralized coordination [102].

The application of agent-based Blockchain Technology in the IoT arena is in its early stage and multifaceted. However, most of the current work deals with security-related topics. The CIoTA framework [103] uses Blockchain Technology to perform distributed and collaborative anomaly detection among resource-constrained IoT devices. Here, software agents runs on every IoT device, and build a local model (an extensible Markov model, precisely) for detecting malicious behaviors in a particular application. Authors of [104–106] focus on SCs. In particular, since centralized access management technologies lack ability to efficiently deal with scalable load, [104] proposes fully distributed access control system for IoT, based on Blockchain Technology and ABC. Aiming at arbitrating roles and permissions, every blockchain node is associated with an agent, which is delegated to the deployment and management of a SC, during the lifetime of the access control system. Similarly, in [106] a MAS exploits SCs to manage the entire supply chain process more efficiently, by automatically writing transaction on the blockchain, verifying that both parties abide to the agreed conditions, and, if these are not met, imposing penalties. In a similar direction, in [105], authors propose to "agentify" the SCs, which currently is based on passive objects, in order to enhance their expressiveness with typical agents' features of autonomy, situatedness, sociality, and intelligence. Going beyond accountability and identity management tasks, there exist works that demonstrate the conceptual and technical feasibility of blockchain-based trustworthy coordination in MAS (see, for instance, [107,108]), i.e., they propose how to ground trustworthy, decentralized coordination in MAS upon blockchain. Nevertheless, these are just preliminary results.

4.6. Semantic technologies

Relationship between agents and semantic technologies (ST) is a complex one. It can be traced back to 2001, when [109], and [110] have been published. The main idea was (and still is) that ST can make agents in more flexible, with easy update of knowledge about the world, and allow agents to understand "the world" deeper and broader. However, already then things were not perfect. For instance, FIPA Ontology Service Specification,[7] which introduced an Ontology Agent (an ontology manager), never became standardized. Furthermore, no popular agent platform implemented such agent (only found reference is [111]). Moreover, developers of one of very popular agent platforms Jade[8] attempted at providing semantic services,[9] using ontologies in agent communication,[10] or combining Jade ontology services

---

[7] http://www.fipa.org/specs/fipa00086.

[8] https://jade.tilab.com.

[9] https://jade.tilab.com/doc/tutorials/clontosupport.pdf.

[10] https://jade.tilab.com/papers/papercaireaart.pdf.

with Protege.[11] However, instead of representing knowledge and forming a flexible knowledge base, ontologies became Java-code fragments.

Overall, most attempts at directly fusing software agents and semantic technologies failed to gain traction. Software agents evolved in three main "non-semantic" directions: (1) general purpose platforms (e.g., Jade), (2) BDI-based platforms (e.g., Jadex or Jason) that, apply Prolog/expert system approach, and (3) simulation platforms (e.g., NetLogo or Repast). At the same time, semantic technologies evolved at their own pace, separately from agent systems.

Nevertheless, one can find a number of projects that attempted at using jointly agent systems and semantics. Let us list a few examples, which illustrate general trends: *(i)* Magenta Technologies: Multi-agent Systems for Ocean logistics – an agent-semantic system solving scheduling problem [112]; (ii) Semantic-based Travel Support System: where semantic representation of "world of travel" is used [113,114]; (iii) Semantic based Workers support system: based on ontologies of organization, travel, research interests, used to support workers in a virtual organization [115–117]; and (iv) Agent-semantic system for management of resources in computational grids (or clouds) [118,119]. Here, an ontology of numerical linear algebra has been developed [120], and possibility of combining agents + semantics + multicriterial analysis has been studied [121].

In all cases, ontologies were stored and processed in a separate repository. In this way, knowledge representation could have been created and updated without (major) changes in the core application/system (see, also, [122]).

The question thus arises, what role can semantic technologies play in the IoT, and how they can be connected with agents. Here, two main scenarios emerge. First, in an IoT ecosystem, a common vocabulary is instantiated. However, we are convinced that (at least for some time to come) only "small", domain specific ontologies will materialize in applications. Here, agents will share common modular ontology [123] and exchange "semantic messages" (ACL messages, with semantic payload) to communicate (see, [124]). Similarly, ontologies can be used to facilitate access control [125,126].

Second, when different data models coexist within a single IoT ecosystem (e.g. due to the ecosystem merger), when interoperability has to be achieved, semantic technologies can be used to translate messages exchanged between "things" [127,128]. This approach can be applied in the ABC. Here, a "translator agent", communicating with individual agents representing each data model, would facilitate the message translation service (see, [129–131]).

In summary, while the past collaboration of agents and semantics was not particularly successful, it seems that both technologies have reached the state when they are ready to join forces to deliver important results in IoT ecosystems.

5. Successful exploitation of agents in IoT applications

Based on paradigms and technologies reported in the previous (sub)sections, the IoT has given rise to several applications in many different fields. In contrast with the mostly theoretical and foundational contributions reported in Section 4, hereinafter some successful agent-based IoT applications and (mostly commercial) related platforms are reported.

The use of SOs and real time analytics to enhance manufacturing and industrial processes, led to the *Industrial IoT (IIoT)* [132]. PTC ThingWorx[12] is a cloud-based end-to-end technology platform (free use is possible, with limited functionality) that enables innovators to rapidly develop and deploy agent-based solutions for the IIoT. The ABC supports the device modeling (different agents are used for the different types of devices), the business logic definition, the design and implementation of collaboration and security mechanisms required for IIoT applications. Recently, PTC ThingWorx included an edge microserver (server software for IoT edge devices) and a software development kit for minimizing devices' power and data demands. Also located in the IIoT arena, Arrayent[13] is a cloud- and agent-based IoT platform specifically designed for manufacturers of mass-market consumer home products. Arrayent empowers brands to get connected and get closer to their customers, by enabling them to connect, monitor and control their IoT products remotely from anywhere. This is possible thanks to lightweight agents, which are deployed on the devices, and have their virtual twins in the Cloud, accessible through RESTful APIs. The ABC helps to manage the devices, their actions and reactions, and their data, as well as in interconnecting different embedded computing platforms (Atmel, Broadcom, Texas Instruments, Raspberry Pi, etc.) and communication protocols (including Wi-Fi, Thread, ZigBee, Z-Wave). Cumulocity IoT[14] is an IoT platform to connect and to experience heterogeneous "things" instantly. It provides pre-integrated devices, or open device agents, and APIs, to turn insight into action using analytics, application integration and workflow management and to, finally, deliver branded services using secure multi-tenancy and role-based access control. Software agents are used as interoperability facilitators, since they translate device-specific interface protocols into reference ones (i.e., REST and JSON) and specific domain models into reference domain ones, and enable secure remote communication across virtually any network. Similarly, CloudPlugs,[15] a container-based, edge to cloud IoT platform for digital transformation and implementation of IIoT initiatives, exploits software agents to eliminate slow and costly firmware development, to quickly develop device applications and to deploy them instantly to thousands of devices. The CloudPlugs agents provide multi-protocol support, secure data ingestion, telemetry, data processing, protocol mediation and local and remote edge control.

The synergy of the IoT and social networking paradigms enables the development of communities where SOs establish humans-like social interactions and temporary collaborations in contexts where expertise and capabilities are spatially distributed. The *Social Internet of Things (SIoT)* aims at simplifying the navigability of a dynamic network of billions of SOs as well as enhancing their efficiency and trustworthiness when providing information and services [40,133]. Speaking Object [134] is a framework pivoting on the argumentation-based coordination: agents can autonomously and dynamically select the better interaction protocol for the current situation, without recurring to prescribed coordination rules. Fostering interoperability in SOs interactions is the goal of agents in iSapiens [135], a framework for designing and implementing smart environments with automatic SOs inclusion leveraging on information such as location, ownership, chronology of mutual interactions. In the Social Factory platform [136] 'twin agents' of humans and SOs are interfaced through broker agents for facilitating their contextualized interactions within a cyberphysical environment across an enterprise social network. Here, the agent-based interfaces allows preventing errors and minimizing out-of-the-loop performance of the human-ware by preserving an adequate level of situation awareness and mental workload. Agents can foster integration of SOs into social networks, for

---

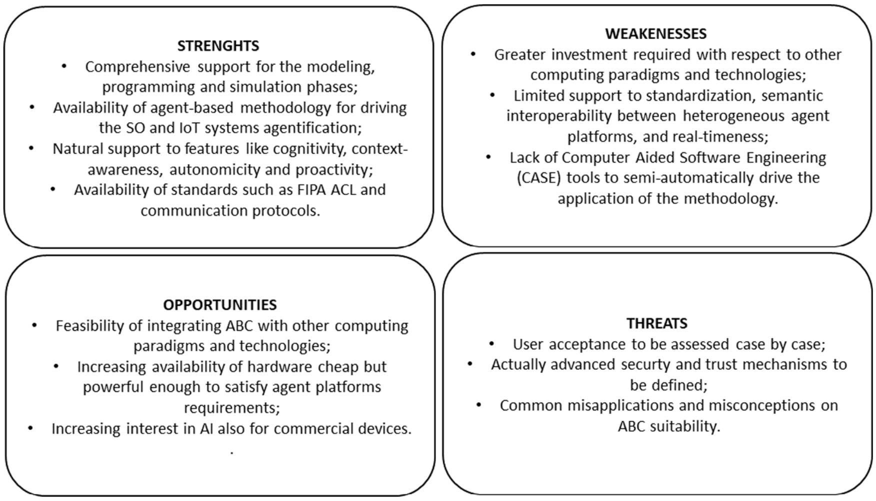
[11] https://jade.tilab.com/doc/tutorials/BeanOntologyTutorial.pdf.

[12] http://www.thingworx.com/.

[13] https://www.arrayent.com/.

[14] https://cumulocity.com/guides/concepts/introduction/.

[15] https://cloudplugs.com/iot-platform-overview/.



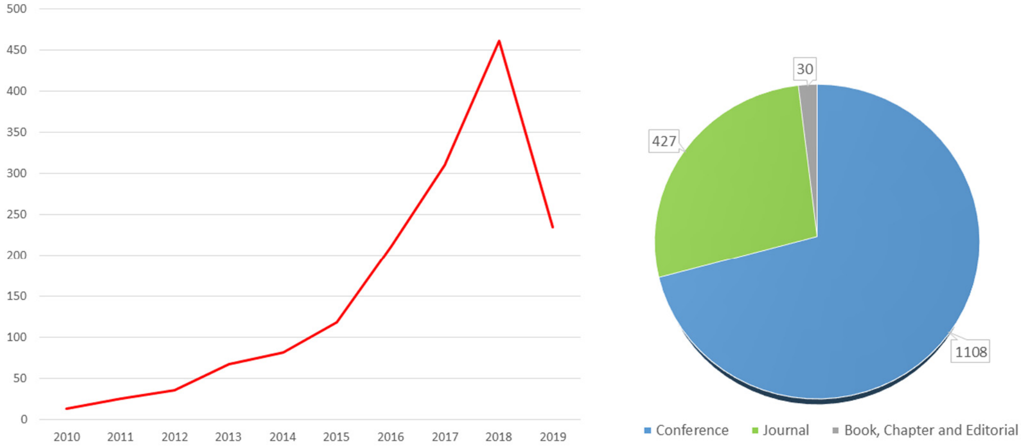
STRENGHTS
- Comprehensive support for the modeling, programming and simulation phases;
- Availability of agent-based methodology for driving the SO and IoT systems agentification;
- Natural support to features like cognitivity, context-awareness, autonomicity and proactivity;
- Availability of standards such as FIPA ACL and communication protocols.

WEAKENESSES
- Greater investment required with respect to other computing paradigms and technologies;
- Limited support to standardization, semantic interoperability between heterogeneous agent platforms, and real-timeness;
- Lack of Computer Aided Software Engineering (CASE) tools to semi-automatically drive the application of the methodology.

OPPORTUNITIES
- Feasibility of integrating ABC with other computing paradigms and technologies;
- Increasing availability of hardware cheap but powerful enough to satisfy agent platforms requirements;
- Increasing interest in AI also for commercial devices.

THREATS
- User acceptance to be assessed case by case;
- Actually advanced securty and trust mechanisms to be defined;
- Common misapplications and misconceptions on ABC suitability.

Fig. 3. SWOT matrix about agent-based IoT development.



(a) (b)

Fig. 4. Agent-based Internet of Things publications (a) per year and (b) per type.

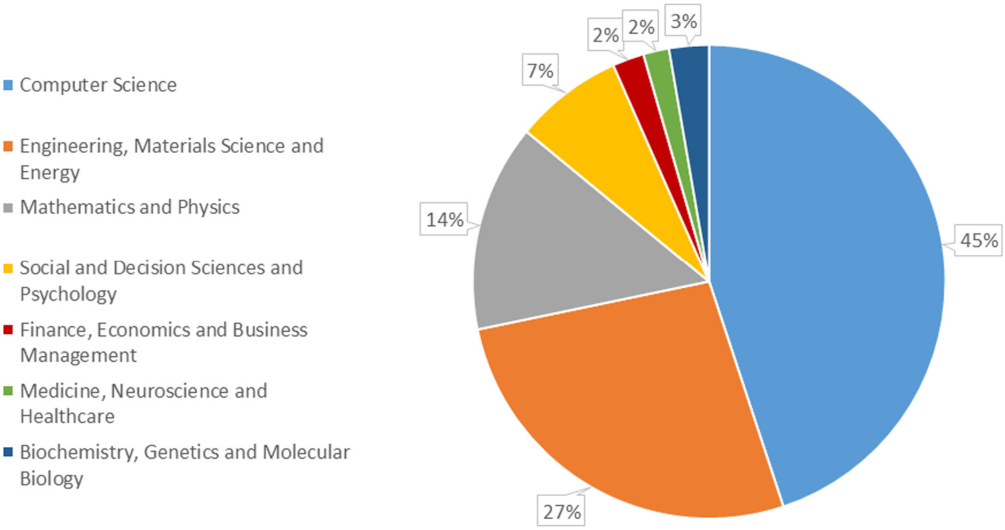
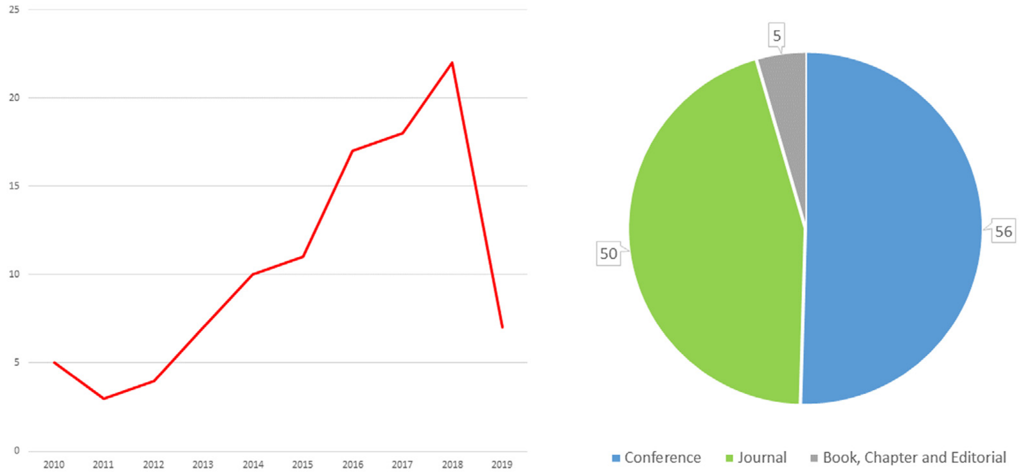


Fig. 5. Agent-based Internet of Things publications per subject area.



(a)

(b)

Fig. 6. Surveyed Agent-based Internet of Things publications (a) per year and (b) per type.

example by updating user's profile, his/her friendships, and by simplifying social network's services discovery and composition [137]. Trustworthiness in SIoT is the focus of [138] and [139]. The former work presents a Trust Framework which promotes social interactions among SOs by associating each of them to a software agent and enabling the decentralized dissemination of their reputation through the Blockchain Technology. The latter, instead, presents a Trust Service Platform, where agents constitute a semi-centralized trust management and reputation system but without exploiting the Blockchain Technology.

The *Internet of Vehicles (IoV)* is a large-scale, distributed system for wireless communication and information exchange between smart vehicles and other SOs (smartphones, smart traffic lights and signals, etc.). As result, complex real-time tasks, such as dynamic and intelligent routing/congestion/traffic management, are performed according to current traffic and environmental conditions [140]. An agent-based approach can be successfully adopted in the IoV scenario because vehicular networks are typically large scale and geographically distributed in dynamic environments, thus requiring autonomous, collaborative and reactive SOs. As reported in [141], several IoV platforms exploit the ABC as modeling and simulation paradigm for complex traffic and transportation systems. For example, [142] proposes JADE-based agents to support traffic operators in determining the best traffic strategies when non-urban roadway meteorological incidents occur. Moving to urban scenarios, TRACK-R [143] is an agent-based platform which exploits a MAS for providing traffic route recommendation to both agentified SOs and humans. Chen et al. [144] developed Mobile-C, a real-time agent-based traffic detection and management system, in which both stationary and mobile agents collaborate in distributed computing and information fusion. Similarly, aiming at an adaptive and smart traffic and transportation control, a highly distributed agent-based traffic management system is reported in [145], in which a sophisticated control algorithm is decomposed on demand, into simple task-oriented agents to adapt to various control scenarios. An agent-based framework for traffic control without traffic signal systems is presented in [146]. Here, each of connected vehicles is modeled as an agent and they all communicate and collaborate through wireless communication technologies for tackling the scalable and flexible problem of intersections. Finally, [147] presents a testbed to experiment and rapidly prototype multi-agent control systems in road traffic management with different strategies. Interestingly, all the aforementioned works conform to the FIPA standards for the sake of interoperability.

6. Concluding analysis and remarks

There are no technology limitations (in terms of computing, storage and communication) hindering the full realization of the IoT ecosystem. However, its multifaceted development issues still need to be comprehensively, and simultaneously, tackled. For example, according to Atzori et al. [2], scalability and self-management used to be separately tackled, in spite of centralized and predefined centralized approaches. This prevented SOs to constitute locally operating, self-organizing and self-adaptive systems.

We believe, and the large number of works analyzed in this survey confirms it, that agents' key features of autonomy, proactivity, intelligence and sociability make the ABC a natural candidate for systematically and effectively developing IoT ecosystems. Indeed, better than other computing paradigms (object-, service-, and component-oriented), the ABC allows modeling both SO/IoT system at different degrees of details and programming (technical, syntactical and semantic) interoperability, autonomicity and distributed intelligence. In addition, the agent-based simulation allows validating multiple design choices before the deployment phase, while agent-based methodologies can systematically and effectively drive both the complete development and the re-engineering process of IoT ecosystem, also in synergy with other paradigms (e.g., cloud and edge computing). However, to ensure that the benefits of an agent-based solution far outweigh its overhead, three pragmatic aspects should be carefully assessed before blindly adopting the ABC paradigm (this holds not only for the IoT, but in any research context). Such aspects respectively deal with technology, economic and conceptual issues.

The *first aspect* refers to relative immaturity of agent technology, which was born and raised not in industry but mainly in academia: as consequence, few agent-based commercial platforms [19] are currently available, and they made no significantly progresses in the last decades [20], particularly regarding standardization, semantic interoperability and real-timeness. The latter is a crucial aspect for many safety-critical IoT applications in which the consequences of, even rare, time failures are potentially disastrous. As underlined by [148], MASs typically adopt best-effort approaches with internal agent schedulers, negotiation protocols, and communication middleware, which do not include comprehensive and global mechanisms for handling real-timeness. This lack affects reliability and predictability.

Moreover, FIPA standards only partially support interoperability in real-time/distributed control and diagnostics [76]. Therefore, extensive interventions in terms of theoretical contributions and practical mechanisms involving the MAS core elements need to be simultaneously and coherently carried out, aiming to real-time compliant agent-based SOs, IoT systems and applications.

The *second aspect* refers to the greater investments required to implement agent-based IoT solutions, typically costlier and less user-friendly than conventional centralized and service-oriented ones [76] (e.g., data flow programming for Web of Things [149]). Indeed, as underlined in [150], agent-based methodologies, tools and languages are designed primarily to serve expert researchers, otherwise requiring consistent learning efforts. For facilitating both non-technical end-users and non-expert researchers in the development of agent-based SOs and IoT systems, frameworks should be easier to understand and use, for example by providing ready-to-use and customizable building blocks and Computer Aided Software Engineering (CASE) tools to semi-automatically drive the application of the methodology.

The *third aspect* refers to, only apparently backdated, misapplications and misconceptions:

- *things can be always profitably agentified*: a large number of dumb devices within the IoT ecosystem (RFID and NFC tags, resource-constrained microcontrollers, etc.) work just with a single thread of control and expose simple conditional behaviors [19,20]. Without complex tasks to be performed (e.g., automatic resolution of policy conflicts, access synchronization to shared resources, dynamic organization without any a-priori network model [151]), such devices should not be designed as agents, which conversely are intrinsically autonomous multi-thread problem solvers;
- *applications can always profitably exploit agents*: it is false that agents are a universal development solution. For example, agent-based techniques fit just the 30% of control tasks and 60% of diagnostic tasks in the industrial scenario [21,76].

Fig. 3 reports and summarizes the main considerations of this analysis under the form of a SWOT (Strengths, Weaknesses, Opportunities, Threats) matrix.

In conclusion, although adopting the ABC paradigm demands for a careful preliminary evaluation of pros and cons, the several described contributions across this survey proved that, to date, an agent-based development approach represents a suitable and effective choice to face the majority of advanced (current and future) SOs and IoT systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by PAS-RAS bilateral project, Poland “Semantic foundation of the Internet of Things”, as well as a collaboration agreement between University of Novi Sad, Serbia, University of Craiova, Romania, SRIPAS, Poland and Warsaw University of Technology, Poland. The works is also partially supported by the PRIN project "Fluidware" N. 2017KRC7KT funded by the MIUR - Italian Government

Appendix

From 2010, year in which the IoT turn on the spotlight, the number of the publications related to the ABC and the IoT has grown steadily. According to Scopus, 1565 works have been published to date, with a marked rise in the last years, as reported in Fig. 4(a). Interestingly, as shown in Fig. 4(b), there are still few books, chapters and editorials on this topic, a good number of journals articles, but many more conference papers. This could be due to the recently risen interest on agent-based IoT: indeed, a general practice is to present preliminary results in workshop/conference papers and to develop it into a journal with new material later on. As matter of fact, the number of published journal articles in the last two years has doubled. Another noteworthy fact concerns the subject area of these publications: as shown in Fig. 5, most of them are (obviously) related to computer science, but also in energy, mathematics, decision science, etc. This confirm the suitability and flexibility of the ABC, perfectly matching with the multifaceted nature of the IoT.

With respect to the publications surveyed in this article (a subset of the available ones, given the wide of the state-of-the-art), we focused on the most relevant ones according to the number of obtained citations, exploitation in/derivation from valuable (inter)national research projects, and personal experience. In particular, as shown in Fig. 6, we covered the whole time windows 2010–2019 and we preferred journal articles, when possible.

References

[1] F. Mattern, C. Floerkemeier, From the internet of computers to the internet of things, in: From Active Data Management to Event-Based Systems and more, Springer, 2010, pp. 242–259.
[2] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (2010) 2787–2805.
[3] P. Patel, D. Cassou, Enabling high-level application development for the internet of things, J. Syst. Softw. 103 (2015) 62–84.
[4] M. Luck, P. McBurney, C. Preist, A manifesto for agent technology: Towards next generation computing, Auton. Agents Multi-Agent Syst. 20 (2004) 3–252.
[5] N.R. Jennings, Agent-based computing: Promise and perils, 1999.
[6] P. Pico-Valencia, J.A. Holgado-Terriza, Agentification of the internet of things: A systematic literature review, Int. J. Distri. Sensor Netw. 14 (10) (2018) 1550147718805945.
[7] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, Agent-based computing in the internet of things: a survey, in: Intl. Symposium on Intelligent and Distributed Computing, Springer, 2017, pp. 307–320.
[8] G. Fortino, W. Russo, C. Savaglio, M. Viroli, M. Zhou, Modeling opportunistic iot services in open iot ecosystems.
[9] B. Molina, C.E. Palau, G. Fortino, A. Guerrieri, C. Savaglio, Empowering smart cities through interoperable sensor network enablers, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 7–12.
[10] G. Fortino, A. Rovella, W. Russo, C. Savaglio, Towards cyberphysical digital libraries: Integrating iot smart objects into digital libraries, in: Management of Cyber Physical Objects in the Future Internet of Things, Springer, 2016, pp. 135–156.
[11] G. Fortino, A. Guerrieri, W. Russo, C. Savaglio, Towards a development methodology for smart object-oriented IoT systems: A metamodel approach, in: Systems, Man, and Cybernetics, 2015 IEEE Intl. Conf. on, IEEE, 2015, pp. 1297–1302.
[12] C. Savaglio, G. Fortino, Autonomic and cognitive architectures for the internet of things, in: Intl. Conf. on Internet and Distributed Computing Systems, Springer, 2015, pp. 39–47.
[13] G. Fortino, W. Russo, C. Savaglio, W. Shen, M. Zhou, Agent-oriented cooperative smart objects: From iot system design to implementation, IEEE Trans. Syst. Man Cybernet.: Syst. 48 (11) (2017) 1939–1956.
[14] A. Ricci, A. Santi, Agent-oriented computing: Agents as a paradigm for computer programming and software development, in: Proc. of the 3rd Int'l Conf. on Future Computational Tech. and Applications, Wilmington: Xpert Publishing Services, Citeseer, 2011, pp. 42–51.
[15] S. Poslad, Specifying protocols for multi-agent systems interaction, ACM Trans. Auton. Adapt. Syst. (TAAS) 2 (15) (2007).

[16] A. Fipa, Fipa acl message structure specification. Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00061/SC00061G.html (30.6.04) 2002.

[17] C.M. Macal, M.J. North, Tutorial on agent-based modeling and simulation, in: Simulation Conf. 2005 Proc. of the Winter, IEEE, 2005, pp. 14–pp.

[18] F. Bergenti, M.-P. Gleizes, F. Zambonelli, Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook, Springer Science & Business Media, 2006.

[19] M.J. Wooldridge, N.R. Jennings, Software engineering with agents: Pitfalls and pratfalls, IEEE Internet Comput. 3 (1999) 20–27.

[20] H.S. Nwana, D.T. Ndumu, A perspective on software agents research, Knowl. Eng. Rev. 14 (1999) 125–142.

[21] M. Gawinecki, M. Kruszyk, M. Paprzycki, M. Ganzha, Pitfalls of agent system development on the basis of a Travel Support System, in: W. Abramowicz (Ed.), Proc. of the BIS 2007 Conf., in: LNCS, vol. 4439, Springer, Berlin, 2007, pp. 488–499.

[22] G. Fortino, W. Russo, C. Savaglio, Agent-oriented modeling and simulation of IoT networks, in: Computer Science and Information Systems (FedCSIS), 2016 Federated Conf. on, IEEE, 2016, pp. 1449–1452.

[23] A. Manzalini, F. Zambonelli, Towards autonomic and situation-aware communication services: the cascadas vision, in: Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on, IEEE, 2006, pp. 383–388.

[24] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A.R. Biswas, K. Moessner, Enabling smart cities through a cognitive management framework for the internet of things, IEEE Commun. Mag. 51 (2013) 102–111.

[25] N.M. do Nascimento, C.J.P. de Lucena, Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things, Inform. Sci. 378 (2017) 161–176.

[26] K. Batool, M.A. Niazi, Modeling the internet of things: a hybrid modeling approach using complex networks and agent-based models, Complex Adapt. Syst. Model. 5 (1) (2017) 4.

[27] M. Dzaferagic, M.M. Butt, M. Murphy, N. Kaminski, N. Marchetti, Agent-based modelling approach for distributed decision support in an iot network, arXiv preprint arXiv:1901.04585, 2019.

[28] C. Savaglio, G. Fortino, M. Zhou, Towards interoperable, cognitive and autonomic iot systems: An agent-based approach, in: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, IEEE, 2016, pp. 58–63.

[29] G. Fortino, A. Guerrieri, W. Russo, Agent-oriented smart objects development, in: Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th Intl. Conf. on, IEEE, 2012, pp. 907–912.

[30] G. Fortino, R. Gravina, W. Russo, C. Savaglio, Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach, Comput. Sci. Eng. 19 (5) (2017) 68–76.

[31] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, V.Y. Terziyan, Smart semantic middleware for the internet of things, ICINCO-ICSO 8 (2008) 169–178.

[32] V. Terziyan, O. Kaykova, D. Zhovtobryukh, Ubiroad: Semantic middleware for context-aware smart road environments, in: Internet and Web Applications and Services (Iciw), 2010 Fifth Intl. Conf. on, IEEE, 2010, pp. 295–302.

[33] T. Leppänen, J. Riekki, M. Liu, E. Harjula, T. Ojala, Mobile agents-based smart objects for the IoT, in: Internet of Things Based on Smart Objects, Springer, 2014, pp. 29–48.

[34] F. Zambonelli, Towards a General Software Engineering Methodology for the Internet of Things. arXiv preprint arXiv:1601.05569, 2016.

[35] A.M. Mzahm, M.S. Ahmad, A.Y. Tang, Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT), in: Intelligent Systems Design and Applications (ISDA), 2013 13th Intl. Conf. on, IEEE, 2013, pp. 159–164.

[36] S. Karnouskos, T.N. De Holanda, Simulation of a smart grid city with software agents, in: Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on, IEEE, 2009, pp. 424–429.

[37] G. D'Angelo, S. Ferretti, V. Ghini, Multi-level simulation of Internet of Things on smart territories, Simul. Model. Pract. Theory 73 (2017) 3–21.

[38] X. Xu, N. Bessis, J. Cao, An autonomic agent trust model for iot systems, Procedia Comput. Sci. 21 (2013) 107–113.

[39] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, K. Long, Cognitive internet of things: a new paradigm beyond connection, IEEE Internet Things J. 1 (2014) 129–143.

[40] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, An edge-based platform for dynamic smart city applications, Future Gener. Comput. Syst. 76 (2017) 106–118.

[41] B. Manate, V.I. Munteanu, T.-F. Fortis, Towards a scalable multi-agent architecture for managing iot data, in: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth Intl. Conf. on, IEEE, 2013, pp. 270–275.

[42] D. Mitrović, M. Ivanović, Z. Budimac, M. Vidaković, Radigost: Interoperable web-based multi-agent platform, J. Syst. Softw. 90 (2014) 167–178.

[43] P. Kasnesis, L. Toumanidis, D. Kogias, C.Z. Patrikakis, I.S. Venieris, ASSIST: An agent-based SIoT simulator, in: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, IEEE, 2016, pp. 353–358.

[44] X. Zhang, R. Adhikari, M. Pipattanasomporn, M. Kuzlu, S.R. Bradley, Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience, in: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, IEEE, 2016, pp. 530–535.

[45] O.S. Kubler, K. Framling, A. Zaslavsky, C. Doukas, E. Olivares, G. Fortino, C.E. Palau, S. Soursos, I. Podnar Åarko, Y. Fang, S. Kr o, C. Heinz, C. Grimm, A. Broering, J. Mitić, K. Olstedt, O. Vermesan, Digitising the Industry: Internet of Things Connecting the Physical, Digital and Virtual Worlds, Vol. 49, River Publishers, 2016, pp. 431–448, chapter 9.

[46] VICINITY - Open virtual neighbourhood network to connect IoT infra-structures and smart objects, http://vicinity2020.eu/vicinity/.

[47] I. Ayala, M. Amor, L. Fuentes, The Sol agent platform: Enabling group communication and interoperability of self-configuring agents in the Internet of Things, J. Ambient Intell. Smart Environ. 7 (2015) 243–269.

[48] M. Ruta, F. Scioscia, G. Loseto, E. Di Sciascio, Semantic-based resource discovery and orchestration in home and building automation: A multi-agent approach, IEEE Trans. Ind. Inform. 10 (2014) 730–741.

[49] T. Kato, R. Chiba, H. Takahashi, T. Kinoshita, Agent-oriented cooperation of iot devices towards advanced logistics, in: Computer Software and Applications Conf. 2015 IEEE 39th Annual, IEEE, 2015, pp. 223–227.

[50] G. Pujolle, An autonomic-oriented architecture for the internet of things, in: Modern Computing, 2006. IEEE John Vincent Atanasoff 2006 Int.l Symp. on, IEEE, 2006, pp. 163–168.

[51] B. Cheng, S. Longo, F. Cirillo, M. Bauer, E. Kovacs, Building a big data platform for smart cities: Experience and lessons from santander, in: Big Data (BigData Congress), 2015 IEEE Intl. Congr. on, IEEE, 2015, pp. 592–599.

[52] J. Kiljander, A. D'elia, F. Morandi, P. Hyttinen, J. Takalo-Mattila, A. Ylisaukko-Oja, J.-P. Soininen, T.S. Cinotti, Semantic interoperability architecture for pervasive computing and internet of things, IEEE Access 2 (2014) 856–873.

[53] A. Tate, Y.-H. Chen-Burger, J. Dalton, S. Potter, D. Richardson, J. Stader, G. Wickler, I. Bankier, C. Walton, P. Williams, I-room: a virtual space for intelligent interaction, IEEE Intell. Syst. 25 (4) (2010) 62–71.

[54] B. Manate, F. Fortis, P. Moore, Applying the prometheus methodology for an internet of things architecture, in: Proc. of the 2014 IEEE/ACM 7th Intl. Conf. on Utility and Cloud Computing, IEEE Computer Society, 2014, pp. 435–442.

[55] N. Spanoudakis, P. Moraitis, Engineering ambient intelligence systems using agent technology, IEEE Intell. Syst. 30 (2015) 60–67.

[56] A. Morris, P. Giorgini, S. Abdel-Naby, Simulating BDI-based wireless sensor networks, in: 2009 IEEE/WIC/ACM Intl. Joint Conf. on Web Intelligence and Intelligent Agent Technology, Volume 2, IEEE, 2009, pp. 78–81.

[57] M. Dyk, A. Najgebauer, D. Pierzchala, Sensesim: An agent-based and discrete event simulator for wireless sensor networks and the internet of things, in: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), IEEE, 2015, pp. 345–350.

[58] S. Berrani, A. Yachir, B. Djemaa, M. Aissani, Extended multi-agent system based service composition in the internet of things, in: 2018 3rd Intl. Conf. on Pattern Analysis and Intelligent Systems (PAIS), IEEE, 2018, pp. 1–8.

[59] P. Krivic, P. Skocir, M. Kusek, G. Jezic, Microservices as agents in iot systems, in: KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Springer, 2017, pp. 22–31.

[60] R.W. Collier, E. O'Neill, D. Lillis, G. O'Hare, Mams: Multi-agent microservicescÅ, in: Companion Proceedings of the 2019 World Wide Web Conference, ACM, 2019, pp. 655–662.

[61] K. Kravari, N. Bassiliades, Storm: A social agent-based trust model for the internet of things adopting microservice architecture, Simul. Model. Pract. Theory 94 (2019) 286–302.

[62] M. Pérez Hernández, B. Alturki, S. Reiff-Marganiec, Fabiot: A flexible agent-based simulation model for iot environments, 2018.

[63] N. Cila, I. Smit, E. Giaccardi, B. Kröse, Products as agents: metaphors for designing the products of the iot age, in: Proc. of the 2017 CHI Conf. on Human Factors in Computing Systems, ACM, 2017, pp. 448–459.

[64] D. Namiot, M. Sneps-Sneppe, On micro-services architecture, Int. J. Open Inf. Technol. 2 (9) (2014) 24–27.

[65] M. Ganzha, M. Paprzycki, W. Pawlowski, P. Szmeja, K. Wasielewska, Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective, J. Netw. Comput. Appl. 81 (2017) 111–124.

[66] G. D'Angelo, S. Ferretti, V. Ghini, Simulation of the internet of things, in: 2016 Intl. Conf. on High Performance Computing & Simulation (HPCS), IEEE, 2016, pp. 1–8.

[67] S. Karnouskos, T.N. De Holanda, Simulation of a smart grid city with software agents, in: 2009 Third UKSim European Symposium on Computer Modeling and Simulation, IEEE, 2009, pp. 424–429.

[68] Q. Zhang, C. Cao, A multi-agent simulation model combined with evolutionary game for cooperative patterns of iot, J. Inf. Comput. Sci. 10 (10) (2013) 2933–2942.

[69] P.C. Ccori, L.C.C. De Biase, M.K. Zuffo, F.S.C. da Silva, Device discovery strategies for the iot, in: 2016 IEEE International Symposium on Consumer Electronics (ISCE), IEEE, 2016, pp. 97–98.

[70] S. Yamaguchi, S. Tsugawa, K. Nakahori, An analysis system of iot services based on agent-oriented petri net pn2, in: 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), IEEE, 2016, pp. 1–2.

[71] H.O. Al-Sakran, Intelligent traffic information system based on integration of internet of things and agent technology, Intl. Journal of Advanced Computer Science and Applications (IJACSA) 6 (2) (2015) 37–43.

[72] G. Fortino, W. Russo, C. Savaglio, Simulation of agent-oriented internet of things systems, in: Proc. 17th Workshop from Objects to Agents, 2016, pp. 8–13.

[73] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, M. Viroli, Modelling and simulation of opportunistic iot services with aggregate computing, Future Gener. Comput. Syst. 91 (2019) 252–262.

[74] S.N. Han, G.M. Lee, N. Crespi, K. Heo, N. Van Luong, M. Brut, P. Gatellier, Dpwsim: A simulation toolkit for iot applications using devices profile for web services, in: 2014 IEEE World Forum on Internet of Things (WF-IoT), IEEE, 2014, pp. 544–547.

[75] T. Jung, N. Jazdi, M. Weyrich, A survey on dynamic simulation of automation systems and components in the internet of things, in: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2017, pp. 1–4.

[76] V. Marik, D. McFarlane, Industrial adoption of agent-based technologies, IEEE Intell. Syst. 20 (2005) 27–35.

[77] M.A. Razzaque, M. Milojevic-Jevric, A. Palade, S. Clarke, Middleware for internet of things: a survey, IEEE Internet Things J. 3 (2016) 70–95.

[78] G. Fortino, A. Guerrieri, W. Russo, C. Savaglio, Middlewares for smart objects and smart environments: overview and comparison, in: Internet of Things Based on Smart Objects, Springer, 2014, pp. 1–27.

[79] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos, Tropos: An agent-oriented software development methodology, Auton. Agents Multi-Agent Syst. 8 (2004) 203–236.

[80] C. Savaglio, W. Russo, G. Fortino, T. Leppänen, J. Riekki, Re-engineering iot systems through acoso-meth: the ietf core based agent framework case study, in: 19th Workshop from Objects to Agents (WOA 2018), Italy, 2018, pp. 28–29.

[81] G. Fortino, A. Guerrieri, W. Russo, C. Savaglio, Integration of agent-based and cloud computing for the smart objects-oriented iot, in: Computer Supported Cooperative Work in Design (CSCWD), Proc. of the 2014 IEEE 18th Intl. Conf. on, IEEE, 2014, pp. 493–498.

[82] G. Fortino, W. Russo, Towards a cloud-assisted and agent-oriented architecture for the internet of things, in: WOA@ AI* IA, Citeseer, 2013, pp. 60–65.

[83] Z. Li, C. Chen, K. Wang, Cloud computing for agent-based urban transportation systems, IEEE Intell. Syst. 26 (1) (2011) 73–79.

[84] A. Singh, D. Juneja, M. Malhotra, Autonomous agent based load balancing algorithm in cloud computing, Procedia Comput. Sci. 45 (2015) 832–841.

[85] K.M. Sim, Agent-based cloud computing, IEEE Trans. Serv. Comput. 5 (4) (2012) 564–577.

[86] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, N. Shiratori, Multiagent-based flexible edge computing architecture for iot, IEEE Netw. 32 (1) (2018) 16–23.

[87] T. Leppänen, C. Savaglio, L. Lovén, W. Russo, G. Di Fatta, J. Riekki, G. Fortino, Developing agent-based smart objects for iot edge computing: Mobile crowdsensing use case, in: Intl. Conf. on Internet and Distributed Computing Systems, Springer, 2018, pp. 235–247.

[88] T. Leppänen, Resource-Oriented Mobile Agent and Software Framework for the Internet of Things, University of Oulu, Faculty of Information Technology and Electrical Engineering, 2018.

[89] F. Cicirelli, A. Guerrieri, A. Mercuri, G. Spezzano, A. Vinci, Itema: A methodological approach for cognitive edge computing iot ecosystems, Future Gener. Comput. Syst. 92 (2019) 189–197.

[90] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, A Cognitive Enabled, Edge-Computing Architecture for Future Generation IoT Environments, in: In the Proceeding of the IEEE 5th World Forum on Internet of Things, Limerick, Ireland, 2019.

[91] V.C. Bumgardner, V.W. Marek, C.D. Hickey, Cresco: A distributed agent-based edge computing framework, in: 2016 12th Intl. Conf. on Network and Service Management (CNSM), IEEE, 2016, pp. 400–405.

[92] F. Aiello, F.L. Bellifemine, 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, Eng. Appl. Artif. Intell. 24 (7) (2011) 1147–1161.

[93] S. Shen, G.M. O'Hare, Wireless.sensor. networks, Wireless sensor networks an energy-aware and utility-based bdi agent approach, Int. J. Sensor Netw. 2 (3/4) (2007) 235–245.

[94] J. Xie, I. Howitt, A. Raja, Cognitive radio resource management using multi-agent systems, in: IEEE CCNC, 2007.

[95] H. Qi, Y. Xu, X. Wang, Mobile-agent-based collaborative signal and information processing in sensor networks, Proc. IEEE 91 (8) (2003) 1172–1183.

[96] M. Chen, T. Kwon, Y. Yuan, V.C. Leung, Mobile agent based wireless sensor networks, J. Comput. 1 (1) (2006) 14–21.

[97] V. Kurbalija, M. Ivanović, Z. Geler, M. Radovanović, Two faces of the framework for analysis and prediction, part 2-research, Inf. Technol. Control 47 (3) (2018) 489–502.

[98] M. Lujak, S. Ossowski, Evacuation route optimization architecture considering human factor, AI Commun. 30 (1) (2017) 53–66.

[99] B. Lorica, How to think about AI and machine learning technologies, and their roles in automation: an overview and framework, including tools that can be used to enable automation, https://www.oreilly.com/ideas/how-to-think-about-ai-and-machine-learning-technologies-and-their-roles-in-automation, (accessed 20.04.19), 2018.

[100] Y. Zhang, A. Grignard, K. Lyons, A. Aubuchon, K. Larson, Real-time machine learning prediction of an agent-based model for urban decision-making, in: Proc. of the 17th Intl. Conf. on Autonomous Agents and MultiAgent Systems, Intl. Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 2171–2173.

[101] N. Nascimento, P. Alencar, C. Lucena, D. Cowan, An iot analytics embodied agent model based on context-aware machine learning, in: 2018 IEEE Intl. Conf. on Big Data (Big Data), IEEE, 2018, pp. 5170–5175.

[102] D. Calvaresi, A. Dubovitskaya, J.P. Calbimonte, K. Taveter, M. Schumacher, Multi-agent systems and blockchain: Results from a systematic literature review, in: Intl. Conf. on Practical Applications of Agents and Multi-Agent Systems, Springer, 2018, pp. 110–126.

[103] T. Golomb, Y. Mirsky, Y. Elovici, Ciota: Collaborative iot anomaly detection via blockchain, arXiv preprint arXiv:1803.03807, 2018.

[104] O. Novo, Blockchain meets iot: An architecture for scalable access management in iot, IEEE Internet Things J. 5 (2) (2018) 1184–1195.

[105] G. Ciatto, A. Maffi, S. Mariani, A. Omicini, Towards agent-oriented blockchains: Autonomous smart contracts, in: PAAMS 2019, 2019.

[106] R. Casado-Vara, J. Prieto, F. De la Prieta, J.M. Corchado, How blockchain improves the supply chain: Case study alimentary supply chain, Procedia Comput. Sci. 134 (2018) 393–398.

[107] G. Ciatto, S. Mariani, A. Omicini, Blockchain for trustworthy coordination: A first study with Linda and Ethereum, in: 2018 IEEE/WIC/ACM Intl. Conf. on Web Intelligence (WI), 2018, pp. 696–703.

[108] E.C. Ferrer, The blockchain: a new framework for robotic swarm systems, in: Proceedings of the Future Technologies Conference, Springer, 2018, pp. 1037–1058.

[109] D. Fensel, Ontologies - a Silver Bullet for Knowledge Management and Electronic Commerce, Springer, 2001.

[110] E. Sirin, B. Parsia, J. Hendler, Template-based composition of semantic web services, in: Agents and the Semantic Web, Papers from the 2005 AAAI Fall Symposium, 2005, pp. 85–92.

[111] P.A. Mitkas, A.L. Symeonidis, D.D. Kehagias, I.N. Athanasiadis, G. Laleci, G. Kurt, Y. Kabak, A.C. Acar, A. Dogac, An agent framework for dynamic agent retraining: Agent academy, CoRR, cs.MA/0407025, 2004.

[112] J. Himoff, Magenta logistics i-scheduler, in: 4th Intl. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2005), July (2005) 25-29, Utrecht, the NetherlandS, 2005, pp. 159–160.

[113] M. Ganzha, M. Gawinecki, M. Paprzycki, R. Gasiorowski, S. Pisarek, W. Hyska, Utilizing semantic web and software agents in a travel support system, in: A.F. Salam, J. Stevens (Eds.), Semantic Web Technologies and EBusiness: Virtual Organization and Business Process Automation, Idea Publishing Group, Hershey, USA, 2006, pp. 325–359.

[114] M. Kruszyk, M. Ganzha, M. Gawinecki, M. Paprzycki, Introducing collaborative filtering into an agent-based travel support system, in: Proc. of the 2007 IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intl. Conf. on Intelligent Agent Technology - Workshops, 2-5 2007, Silicon Valley, CA, USA, 2007, pp. 439–443.

[115] S.K. Rhee, J. Lee, M. Park, M. Szymczak, G. Frackowiak, M. Ganzha, M. Paprzycki, Measuring semantic closeness of ontologically demarcated resources, Fund. Inform. 96 (4) (2009) 395–418.

[116] G. Frackowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, M. Park, Y. Han, Considering resource management in agent-based virtual organization, in: Intelligent Agents in the Evolution of Web and Applications, 2009, pp. 161–190.

[117] M. Szymczak, G. Frackowiak, M. Gawinecki, M. Ganzha, M. Paprzycki, M. Park, Y. Han, Y.T. Sohn, Adaptive information provisioning in an agent-based virtual organization-ontologies in the system, in: Agent and Multi-Agent Systems: Technologies and Applications, Second KES Intl. Symposium, KES-AMSTA 2008, Incheon, Korea, March (2008) 26-28. Proc., 2008, pp. 1271–280.

[118] M. Paprzycki, M. Drozdowicz, M. Ganzha, K. Wasielewska, I. Lirkov, R. Olejnik, N. Attaoui, Utilization of modified coregrid ontology in an agent-based grid resource management system, in: Proc. of the ISCA 25th Intl. Conf. on Computers and Their Applications, CATA 2010, March (2010) 24-26, Sheraton Waikiki Hotel, Honolulu, Hawaii, USA, 2010, pp. 240–245.
[119] K. Wasielewska, M. Ganzha, M. Paprzycki, C. Badica, M. Ivanovic, I. Lirkov, S. Fidanova, Agents in grid extended to clouds, in AIP Conf. Proc. Vol. 1773, 2016.
[120] K. Wasielewska, M. Ganzha, M. Paprzycki, I. Lirkov, Developing ontological model of computational linear algebra – preliminary considerations, in: AIP Conf. Proc. Vol. 1561, 2013, pp. 133–143.
[121] K. Wasielewska, M. Ganzha, M. Paprzycki, C. Badica, M. Ivanovic, I. Lirkov, Multicriteria analysis of ontologically represented information, in: AIP Conf. Proc. Vol. 1629, 2014, pp. 281–295.
[122] G. Frackowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, C. Badica, Y. Han, M. Park, Adaptability in an agent-based virtual organization, IJAOSE 3 (2/3) (2009) 188–211.
[123] M. Ganzha, M.M. Mesjasz, M. Paprzycki, M. Ouedraogo, Inserting brains into software agents - preliminary considerations, in: Internet and Distributed Computing Systems - 7th Intl. Conf. IDCS 2014, Calabria, Italy, September (2014) 22-24. Proc., 2014, pp. 3–14.
[124] M.M. Mesjasz, D. Cimadoro, S. Galzarano, M. Ganzha, G. Fortino, M. Paprzycki, Integrating jade and maps for the development of agent-based wsn applications, in: Intelligent Distributed Computing VI - Proc. of the 6th Intl. Symposium on Intelligent Distributed Computing - IDC 2012, Calabria, Italy, 2012, 2012, pp. 211–220.
[125] M. Drozdowicz, M. Alwazir, M. Ganzha, M. Paprzycki, Graphical interface for ontology mapping with application to access control, in: Intelligent Information and Database Systems - 9th Asian Conf., ACIIDS 2017, Kanazawa, Japan, April (2017) 3-5, Proc., Part I, 2017, pp. 46–55.
[126] M. Drozdowicz, M. Ganzha, M. Paprzycki, Semantic policy information point - preliminary considerations, in: ICT Innovations 2015 - Emerging Technologies for Better Living, Ohrid, Macedonia, 1-4 October, 2015, 2015, pp. 11–19.
[127] M. Ganzha, M. Paprzycki, W. Pawlowski, P. Szmeja, K. Wasielewska, Semantic interoperability in the internet of things: An overview from the inter-iot perspective, J. Netw. Comput. Appl. 81 (2017) 111–124, [Online] Available: https://doi.org/101016/j.jnca.201608007.
[128] M. Ganzha, M. Paprzycki, W. Pawowski, P. Szmeja, K. Wasielewska, Alignment-based semantic translation of geospatial data, in: 2017 3rd Intl. Conf. on Advances in Computing, Communication Automation (ICACCA) (Fall), 2017, pp. 1–8.
[129] R. Gonzalez-Usach, C.E. Palau, M. Julian, A. Belsa, M.A. Llorente, M. Montesinos, M. Ganzha, K. Wasielewska, P. Sala, Use cases, applications and implementation aspects for iot interoperability, in: O. Vermesan, J. Bacquet (Eds.), Distributed Intelligence at the Edge and Human Machine-To-Machine Cooperation, River Publishers, 2018, pp. 139–173.
[130] J.L.R. Moreira, L. Daniele, L.F. Pires, M. van Sinderen, K. Wasielewska, P. Szmeja, W. Pawlowski, M. Ganzha, M. Paprzycki, Towards iot platforms' integration semantic translations between W3C SSN and ETSI SAREF, in Joint Proc. of SEMANTiCS 2017 Workshops co-located with the 13th Intl. Conf. on Semantic Systems (SEMANTiCS 2017), Amsterdam, Netherlands, September 11 and 14, 2017. 2017.
[131] M. Ganzha, M. Paprzycki, W. Paw owski, P. Szmeja, K. Wasielewska, Streaming semantic translations, in: 2017 21st Intl. Conf. on System Theory, Control and Computing (ICSTCC), 2017, pp. 1–8.
[132] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination, Comput. Netw. 101 (2016) 158–168.
[133] H.L. Nguyen, J.E. Jung, Socioscope: A framework for understanding internet of social knowledge, Future Gener. Comput. Syst. 83 (2018) 358–365.
[134] M. Lippi, M. Mamei, S. Mariani, F. Zambonelli, An argumentation-based perspective over the social iot, IEEE Internet Things J. 5 (4) (2018) 2537–2547.
[135] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, O. Briante, G. Ruggeri, Isapiens: a platform for social and pervasive smart environments, in: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), IEEE, 2016, pp. 365–370.
[136] D. Romero, T. Wuest, J. Stahre, D. Gorecky, Social factory architecture: social networking services and production scenarios through the social internet of things, services and people for the social operator 4.0, in: IFIP Intl. Conf. on Advances in Production Management Systems, Springer, 2017, pp. 265–273.
[137] L. Atzori, A. Iera, G. Morabito, M. Nitti, The social internet of things (siot)– when social networks meet the internet of things: Concept, architecture and network characterization, Comput. Netw. 56 (16) (2012) 3594–3608.
[138] G. Fortino, F. Messina, D. Rosaci, G.M. Sarne, Using blockchain in a reputation-based model for grouping agents in the internet of things, IEEE Trans. Eng. Manage. (2019).
[139] N.B. Truong, T.-W. Um, G.M. Lee, A reputation and knowledge based trust service platform for trustworthy social internet of things, in: Innovations in Clouds, Internet and Networks (ICIN), 2016.
[140] M. Chen, Y. Tian, G. Fortino, J. Zhang, I. Humar, Cognitive internet of vehicles, Comput. Commun. 120 (2018) 58–70.
[141] B. Chen, H.H. Cheng, A review of the applications of agent technology in traffic and transportation systems, IEEE Trans. Intell. Transp. Syst. 11 (2) (2010) 485–497.
[142] V.R. Tomás, L.A. García, Agent-based management of non urban road meteorological incidents, in: Intl. Central and Eastern European Conf. on Multi-Agent Systems, Springer, 2005, pp. 213–222.
[143] A. Garcia-Serrano, D.T. Vioque, F. Carbone, V. Mendez, Fipa-compliant mas development for road traffic management with a knowledge-based approach: The track-r agents, in: Proc. Challenges Open Agent Syst. Workshop, 2003.
[144] B. Chen, H.H. Cheng, J. Palen, Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems, Transp. Res. C 17 (1) (2009) 1–10.
[145] F.-Y. Wang, Toward a revolution in transportation operations: Ai for complex systems, IEEE Intell. Syst. 23 (6) (2008) 8–13.
[146] K.-H.N. Bui, J.J. Jung, Internet of agents framework for connected vehicles: A case study on distributed traffic control system, J. Parallel Distrib. Comput. 116 (2018) 89–95.
[147] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, J. Hellendoorn, A test bed for multi-agent control systems in road traffic management, in: Applications of Agent Technology in Traffic and Transportation, Springer, 2005, pp. 113–131.
[148] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, G. Buttazzo, The challenge of real-time multi-agent systems for enabling iot and cps, in: Proc. of the Intl. Conf. on Web Intelligence, ACM, 2017, pp. 356–364.
[149] D. Guinard, V. Trifa, T. Pham, O. Liechti, Towards physical mashups in the web of things, in: Networked Sensing Systems (INSS), 2009 Sixth Intl. Conf. on, IEEE, 2009, pp. 1–4.
[150] H. Yu, Z. Shen, C. Leung, From internet of things to internet of agents, in: 2013 IEEE Intl. Conf. on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, IEEE, 2013, pp. 1054–1057.
[151] C. Savaglio, P. Pace, G. Aloi, A. Liotta, G. Fortino, Lightweight reinforcement learning for energy efficient communications in wireless sensor networks, IEEE Access 7 (2019) 29 355–29 364.

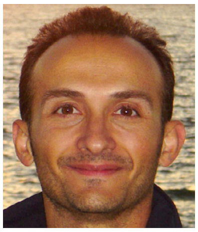
Claudio Savaglio received his B.S., M.S. and Ph.D. degrees in Computer Engineering respectively in 2010, 2013 and 2018 from the University of Calabria, where he is currently serving as PostDoctoral Researcher. In 2013 he was Visiting Researcher at University of Texas at Dallas (TX, U.S.A.), in 2016 at New Jersey Institute of Technology, (NJ, U.S.A.), and in 2017 at Universitat Politecnica de Valencia (Valencia, Spain). His research interests include the Internet of Things, Edge Computing, network simulation, and agent-oriented middleware and development methodologies.

Maria Ganzha is an Associate Professor in the Warsaw University of Technology (Warsaw, Poland). She got an MS and a Ph.D. degrees in mathematics from the Moscow State University, Russia, and a Doctor of Science degree in Computer Science from the Polish Academy of Sciences. Maria has published closed to 200 research papers, is on editorial boards of 5 journals and a book series, and was invited to program committees of more than 250 conferences. She was also the Principal Investigator, of the SRIPAS team, in the INTER-IoT project. Here, her team was responsible for use of semantic technologies in the context of interoperability of IoT platforms.

Marcin Paprzycki is an associate professor at the Systems Research Institute, Polish Academy of Sciences. He has an MS from Adam Mickiewicz University in Poznań, Poland, a Ph.D. from Southern Methodist University in Dallas, Texas, and a Doctor of Science from the Bulgarian Academy of Sciences. He is a senior member of IEEE, a senior member of ACM, a Senior Fulbright Lecturer, and an IEEE CS Distinguished Visitor. He has contributed to more than 450 publications and was invited to the program committees of over 500 international conferences. He is on the editorial boards of 12 journals.

Costin Bădică is Professor at the Department of Computers and Information Technology, Faculty of Automation, Computers and Electronics, University of Craiova, Romania since 2006. He obtained the Ph.D. in Automatic Control in 1999. The research of Prof. Costin Bădică is at the intersection of Artificial Intelligence, Software Engineering and Distributed Systems. He obtained results in applied formal representations and reasoning, as well as intelligent distributed multi-agent systems, platforms and programming. Prof. Costin Bădică co-initiated the "Intelligent Distributed Computing — IDC" series of conferences. Prof. Costin Bădică is Founding Member of Romanian Association of Artificial Intelligence. He organized as Guest Editor many international journal special issues and he managed many national and international research projects.

Mirjana Ivanović Ph.D. Since 2002 holds position of full professor at Faculty of Sciences, University of Novi Sad, Serbia. She is member of University Council for informatics for more than 10 years. Author or co-author is, of 13 textbooks, 13 edited proceedings, 3 monographs, and of more than 440 research papers on multi-agent systems, e-learning and web-based learning, applications of intelligent techniques (CBR, data and web mining), software engineering education, and most of which are published in international journals and proceedings of high-quality international conferences. She is/was a member of Program Committees of more than 200 international Conferences and General Chair and Program Committee Chair of numerous international conferences. Also she has been invited speaker at several international conferences and visiting lecturer in Australia, Thailand and China. As leader and researcher she has been participated in numerous international projects. Currently she is Editor-in-Chief of Computer Science and Information Systems Journal.



Giancarlo Fortino (SM'12) is Professor of Computer Engineering at the Dept. of Informatics, Modeling, Electronics, and Systems of the University of Calabria, Italy. He received a Ph.D. in Computer Engineering from Unical 2000. His research interests include agent-based computing, wireless sensor networks, IoT technology. He is author of over 300 papers in international journals, conferences and books. He is co-founder and CEO of SenSysCal S.r.l., a Unical spin-off focused on innovative IoT systems.