A Scalable Agent-Based Smart Environment for Edge-Based Urban IoT Systems

Franco Cicirelli[1], Giancarlo Fortino[1,2], Antonio Guerrieri[1], Giandomenico Spezzano[1(✉)], and Andrea Vinci[1]

[1] CNR – National Research Council of Italy, Institute for High Performance Computing and Networking (ICAR), Via P. Bucci 7-11C, 87036 Rende (CS), Italy
{cicirelli,guerrieri,spezzano,vinci}@icar.cnr.it

[2] Department of Informatics, Modeling, Electronics, and Systems, University of Calabria, 87036 Rende, Italy
fortino@unical.it

Abstract. New Internet of Things (IoT) applications are encouraging Smart City and Smart Environments initiatives all over the world, by leveraging big data and ubiquitous connectivity. This new technology enables systems to monitor, manage and control devices, and to create new knowledge and actionable information, by the real-time analysis of data streams. In order to develop applications in the depicted scenario, the adoption of new paradigms is required. This paper suggests combining the emergent concept of edge/fog computing with the agent metaphor, so as to enable designing systems based on the decentralization of control functions over distributed autonomous and cooperative entities, which run at the edge of the network. Moreover, we suggest the adoption of the *iSapiens* platform as a reference, as it was designed specifically for the mentioned purposes. Multi-agent applications running on top of *iSapiens* can create smart services using adaptive and decentralized algorithms which exploit the principles of cognitive IoT.

Keywords: IoT · Urban computing · Smart environments
Intelligent agents · Edge computing

1 Introduction

The Internet of Things is realizing a scenario where interconnected objects or things can behave as autonomous entities and can cooperate among them for reaching some common goal, with minimum human intervention. Such smart things must have the ability to sense the environment where they are immersed, analyze the collected information, and take decisions and actions to achieve their objectives.

Many IoT applications are not yet developed because they require being capable of scaling to incorporate a huge number of objects [1]. Centralized solutions

G. Fortino et al. (Eds.): InterIoT 2017/SaSeIoT 2017, LNICST 242, pp. 53–59, 2018.
https://doi.org/10.1007/978-3-319-93797-7_7

do not fit well with such requirement, and systems that have static configuration are not adequate for managing the complexity of environments which evolve continuously. Several IoT applications rely on a cloud-based architecture, where the main computation and storage is concentrated on remote servers. Such approach is not capable of realizing application having real-time requirements, high volume of data, or limited network bandwidth constraints. This raises the need for approaches relying on decentralized processing, where mission-critical processes can run locally even if communication with the operation centers is not adequate. An architecture capable of distributing computation in edge nodes is often the only exploitable solution. However, in order to exploit edge computing for realizing IoT applications, many challenges must be addressed.

This paper proposes an agent-oriented framework called *iSapiens* [2] for developing IoT smart environments based on a networks of smart objects, which can be exploited for the creation of services related to transportation, parking, lighting, traffic, waste and safety [3]. In the proposed framework, smart objects are modeled as agents, running on a multi-agent system, which cooperate for achieving specific goals. We propose to use agents, running close to the controlled objects, in order to mitigate the issues of lack of scalability and reliability, and to furnish reasoning and intelligence capabilities to the things part of an IoT system. The idea is to mirror every thing with an agent, having embedded reasoning, intelligence, and cooperation capabilities. The ability of objects to reason about the environments where they are immersed can contribute to useful outcomes for humans, using cognitive IoT techniques [4], and through the exploitation of self-organizing, decentralized control mechanisms.

This vision is made possible using the concept of *edge computing*. The term edge computing, also referred to as *fog computing*, essentially means that, instead of working from a remote cloud, systems operate on network ends [5]. Thus, data can be processed closer to smart devices rather than being sent to the cloud for elaboration. This approach well suits with IoT because it allows action in real time on the incoming data, without exceeding limits of available bandwidth. By using this distributed strategy, it is possible to both lower costs and improve efficiency.

The paper is structured as follows: Sect. 2 presents the cognitive model to develop a self-configuring agents. Section 3 describes the three-layer software architecture of the platform *iSapiens*. Section 4 presents the use of *iSapiens* platform for potential applications in the surveillance sector using video/audio analytics techniques. Finally, Sect. 5 concludes the paper.

2 Self-adaptive Embodied Agents

Networked objects with self-configuration mechanisms make it possible for distributed systems to adapt to dynamic environments by automatically bootstrapping and reconfiguring themselves which is part of the self-managed process of an automatic system. In this context, Neto et al. [6] implemented a platform that helps the realization of self-adaptive agents with the aim of realizing *autonomic computing*, as proposed by IBM.

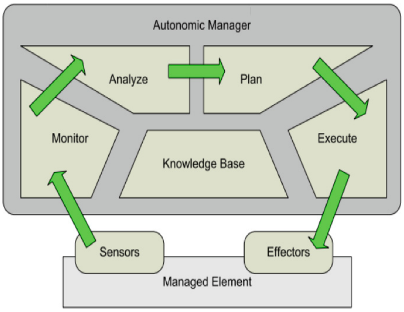

Fig. 1. Autonomic cycle.

Such adaptive agents rely on a loop of control (see Fig. 1) consisting in four activities, namely:

- **Collect**: realizes the collection of the application data;
- **Analyze**: realizes the analysis of the collected data to try in detecting if there are problems;
- **Plan**: realizes the decisions on what to do in the case in which problems are detected;
- **Execute**: realizes some changes in the run of the application, consequently to the executed actions.

The control loop coming from the proposal by IBM and that defines the behavior of the self-adaptive agents can be customized. Moreover, instead of running analysis and planning activities, the adaptive agents can realize decisions by exploiting *controllers* data. Such controllers could be defined as an intelligent entity able to adapt relationships with users and the environment. As the environment is constantly evolving in time and connecting to the network by using sensors and actuators. The autonomic computing approach can be considered as appropriate to design the controller enabling the adaptivity and reconfigurability of the system.

We have defined a controller as an intelligent agent that is capable of sensing the environment, infer knowledge from data and generalize what has already learned in the future situations. The common shared model useful for the description of the behavior of an intelligent (cognitive) agent is the so-called cognitive cycle (see Fig. 2) which is usually considered as composed by four phases:

- **Sensing**: when the system continuously acquires information and knowledge about the objects that interact in the system and about their internal statuses;
- **Analysis**: here the gathered knowledge is elaborated so to obtain a peculiar and concise awareness;
- **Decision**: here, the processed data from the analysis phase is elaborated and is selected a new system configuration according to the given objectives;
- **Action**: in this state, the system runs the configuration that have been provided by the Decision phase.

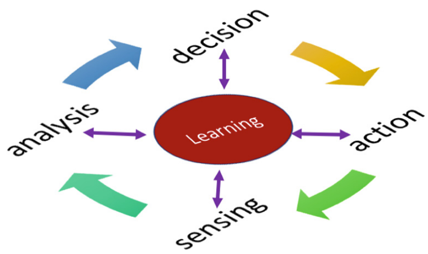

Fig. 2. Cognitive cycle.

In this schema, the learning phase involves all the stages (within certain limits) of the cognitive cycle, thus being continuously executed.

The cognitive cycle introduced above can be mapped straightforwardly to agents' behaviors.

In creating an intelligent agent, the sensing and actuation functions will be inherent in the object; for example, when constructing an intelligent lamp, the bulb (actuator) will be part of the lamp. However, the computational capabilities of the object may follow on of these patterns:

- **Embedded computation**, in which a processor, memory and storage are built into the object
- **Augmented computation**, in which a computational device is connected to an everyday object (e.g. an intelligent chair can be developed by adding pressure sensors and an Arduino controller to an existing office chair)
- **External computation**, in which the object perform computation by communicating via a network interface with an agent running in a edge node.

3 iSapiens Architecture

iSapiens is a Java-based platform for the development of a pervasive SE. A typical *iSapiens* application comprehends a set of physical devices (which can be sensors, actuators or even more complex smart objects) directly spread in a smart environment and connected to a network of *iSapiens* computing nodes (see Fig. 3).

The agent metaphor is used for designing and implementing the business logic of the application to develop. Software agents execute on the *iSapiens* computational nodes close to the physical devices they have to control, thus implementing edge computation (in-network computation). On behalf of the final users, agents can also exploit out-of-the-edge services (off-network computation) which can be purposely developed for a specific application, or which can be made directly available by third-party service providers. Off-network computation is useful when an application requires specific functions that cannot be executed on the edge nodes due, e.g., to computational limitations, or in the

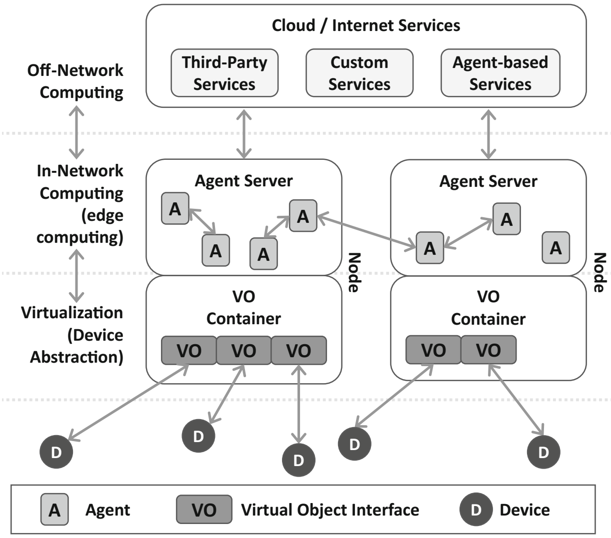

Fig. 3. iSapiens's architecture.

case that a centralized and global view of the system is required. Such functions include predictive analysis, data mining, off-line analytics, and so on.

The devices connected to an iSapiens node are abstracted as the union of Virtual Objects (VOs). All the VOs residing in a specific node are managed by a Virtual Objects Container. VOs offer a transparent access to the physical part due to a common and well-established interface exposed as API. The VO abstraction aims at hiding the low level heterogeneity in terms of communication protocols and specific hardware of all the connected physical devices.

Each *iSapiens* computing node also hosts an Agent Server that is responsible for the management, the execution, and for supporting communication among software agents. Communication is based on asynchronous exchange of messages. The platform offers a specific kind of messages, namely acquaintance messages, that can be used to dynamically establish asymmetric acquaintance relationships among agents. An agent can use the hardware devices abstracted through VOs, and it can access the social network of a controlled device through the SO representing the device from a social perspective. Social-based services are accessed by functionalities provides from the Yellow Pages Agents. The Yellow Pages agents, all-together, implement a distributed service which permits the search of previously registered applicative agents on the basis of attributes specified as <key, value> pairs. Agent Servers and VO Containers running on each networked *iSapiens* computing node, constitute a middleware allowing the exploitation of distributed in-network computation. All the components introduced so far, either a physical device, a virtual object, an agent, or a whole computing node, can be dynamically added, removed, or up-dated. Also the relationships among agents can be dynamically updated through yellow pages and acquaintance messages. All of this allows the development of scalable, extensible and pervasive applications.

The above described architecture is fully compliant with the paradigm of edge computing. A way to look at edge computing is to consider it as a virtualized platform that is typically located between end user devices and the cloud data centers hosted within the Internet. Thus edge computing can provide better quality of service in terms of delay, power consumption, reduced data traffic over the Internet. The main feature of edge computing is its ability to support applications that require low latency, location awareness and mobility. This ability is made possible by the fact that the edge computing systems are deployed very close to the end users in a widely distributed manner. Edge computing nodes hosted must possess sufficient computing power and storage capacity to handle the resource intensive user requests.

4 Potential Applications

The need of protecting infrastructures is an important task that sometimes goes beyond predictive maintenance.

iSapiens actually have been used in protecting infrastructures such as streets, school buildings, water supply, sewer through deployments of cameras and by combining video analytics with other devices like motion sensors, radar or acoustic/seismic sensors. In this way, probability of detection increases dramatically with a reciprocal decrease in false alarm rates [7].

In the context of video surveillance, detecting anomalies is of paramount importance. In this field, iSapiens detects anomalies by using a novel mechanism able to detect if normal or abnormal things happen over time. We have defined some anomaly indexes that, when overcome, represent an anomaly [8,9].

These technologies are very useful for security and safety purposes. Video analytics is also used to detect anomalies in the density of people at some events.

Nowadays, video surveillance systems can be deployed to store what happens in an area of interest. These systems take signals that are related to the security and to the safety. Video analytics systems enhance such video surveillance systems by adding real time detection on events together with analysis of what is already happened (the event). All of this is done to increase the effectiveness of the video surveillance system with respect to the traditional one. In this field, audio analytics is added to have more information about the controlled place. Moreover, not everything can be taken by some cameras. In this context, our audio analytics can be able to detect dangers.

Building a system capable of recognizing and reacting to the sounds around us means, at the best of our knowledge, doing things that have never been done before.

We envision a future where omnipresent, intelligent, context-aware computing is able to better help people by responding to the sounds around them, no matter where they are.

5 Conclusions

We have proposed a solution to support a new approach to mitigate the issues of lack of scalability, reliability, reasoning and intelligence in things in the IoT systems. *iSapiens* an agent-based framework has been presented to developing analytics for embedded systems in an environment of fog computing. In many case, the analytics run directly in embodied agents allocated in the sensors. Potential applications that we are developing through *iSapiens* are mentioned and discussed.

Acknowledgements. This work has been partially supported by "*Smart platform for monitoring and management of in-home security and safety of people and structures*" project that is part of the DOMUS District, funded by the Italian Government (PON03PE_00050_1).

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**, 2787–2805 (2010)
2. Cicirelli, F., Guerrieri, A., Spezzano, G., Vinci, A., Briante, O., Ruggeri, G.: iSapiens: a platform for social and pervasive smart environments. In: WF-IoT, pp. 365–370 (2016)
3. Roukounaki, A., Soldatos, J., Petrolo, R., Loscri, V., Mitton, N., Serrano, M.: Visual development environment for semantically interoperable smart cities applications. In: Mandler, B., et al. (eds.) IoT360 2015. LNICST, vol. 170, pp. 439–449. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47075-7_48
4. Liotta, A.: The cognitive net is coming. IEEE Spectr. **50**(8), 26–31 (2013)
5. Bonomi, F., Milito, R., Natarajan, P., Zhu, J.: Fog computing: a platform for Internet of Things and analytics. In: Bessis, N., Dobre, C. (eds.) Big Data and Internet of Things: A Roadmap for Smart Environments. SCI, vol. 546, pp. 169–186. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05029-4_7
6. Neto, B., Costa, A., Netto, M., Silva, V., Lucena, C.: JAAF: a framework to implement self-adaptive agents. In: International Conference on Software Engineering and Knowledge Engineering (2009)
7. Cicirelli, F., Guerrieri, A., Spezzano, G., Vinci, A.: An edge-based platform for dynamic Smart City applications. Future Gener. Comp. Syst. **76**, 106–118 (2017)
8. Bosman, H., Iacca, G., Tejada, A., Wortche, H., Liotta, A.: Spatial anomaly detection in sensor networks using neighborhood information. Inf. Fusion J. **33**, 41–56 (2017)
9. Bosman, H., Iacca, G., Tejada, A., Wortche, H., Liotta, A.: Ensembles of incremental learners to detect anomalies in ad-hoc sensor networks. Ad Hoc Netw. (Special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies) **35**, 14–36 (2015)