Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 416–423

Procedia Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018)

CityMii - An integration and interoperable middleware to manage a Smart City

José Cecílio [a,b]*, Filipe Caldeira[c], Cristina Wanzeller[c]

[a]University of Lisbon, Lisbon, Portugal
[b]University of Coimbra, Coimbra, Portugal
[c]Polytechnic Institute of Viseu, Viseu, Portugal

Abstract

Modern cities are supported by multiple heterogeneous IT systems deployed and managed by distinct agents. In general, those systems use old, dependent and non-standardized technologies, which make them legacy and incompatible systems. As smart cities are moving toward a fully centralized management approach, the lack of integration among systems raises several problems. Since they are independent, it is not easy to correlate information from different systems and put it together to work in order to achieve application goals.

The collaboration among different systems enables an agent to offer new functionalities (services or just information about the city) that cannot be provided by any of these systems working as individual entities. The goal of this paper is to propose an integration middleware to support the management of Smart Cities in a dynamic, transparent and scalable way. The proposed middleware intends to support interoperability among different systems operating in a city.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Smart Cities, Middleware, Integration, Interoperability;

1. Introduction

Nowadays, cities rely on multiple systems (e.g. smart grid, efficient buildings, healthcare, garbage, water) which make it smarter. However, typically those systems are from different providers and are managed by distinct agents, using their own computational infrastructure, and work in isolation. In addition, those systems are often incompatible since in general they use old, dependent and non-standardized technologies. This results in an

* Corresponding author. Tel.: +351 217 500 087.
E-mail address: jmcecilio@fc.ul.pt

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.062

environment in which there is no interoperability among the systems, debilitating the goal of fully addressing the urban development challenges.

An intelligent city need tools able to contribute to an efficient management and orchestration of the existing different services. It is important to design and implement solutions for urban management based on the knowledge of the state of the city (in each instance – real time or near it) allowing to share information with third-party services thus promoting quality of life within the city.

The information of the city (sensed data) is collected through sensors distributed over urban space, which communicate using wired or wireless networks. These sensors may be owned by cities, but may also be owned by different stakeholders that supply services on the city infrastructure. Typically, the data is collected and handled by different management systems, owned and managed by each stakeholder and do not integrate with others.

The goal of this paper is to propose a framework that allows city authorities to efficiently manage the city by centralizing all relevant data in a single point while enabling to gather knowledge from correlated data. This will allow to process data collected by different systems, presenting it in a structured way that enables extracting new insights from this data in order to facilitate the decision-making process. This framework aims at supporting interoperability among the different systems of the city by providing a set of services and tools to have direct control over the environment, allowing a real-time update of the different devices distributed in the urban space. Once developed, the proposed solution will enable developers and city authorities to provide richer and more interesting applications to help them solving problems detected on the city ultimately providing better services to citizens.

2. Related work

Smart cities and IoT applications have received a considerable attention from researchers in recent years. Smart cities can provide a new generation of real-time and time-critical, location, social, and context-aware services to their digital citizens, such as for emergency and health-care [1], surveillance [2], entertainment, and social good [3, 4]. Recent research activity related with smart cities, focuses on event forecasting [5], multi-sensor information fusion [6], business model and profit maximization [7], ontologies [8], service models [9] and quality of experience [10]. Researchers have also developed a multitude of application-specific solutions for areas such as diagnostics [11,12], environmental monitoring [14, 15], social interest [16], traffic management [17], etc. However, these application-specific solutions are not integrated.

The lack of integration raises some problems, for example, how distinct, heterogeneous, fully decentralized and independent systems can closely collaborate with each other to achieve innovative application and to manage a city from a single point.

In the literature, there is a set of works proposing middleware approaches [18, 19, 20, 21, 22] which provide abstractions to develop Smart City applications. Nevertheless, in general, they do not fully address the interoperability needed to manage a real city. For instance, some of them are focused on IoT environments and in other abstractions for Smart City [18, 19, 20]. Other middleware approaches focuses on the integration of public and private cloud platforms for smart cities environments [21, 22] rather than interoperability. In contrast, OverStar [23] does have a strong focus on interoperability but its generality fails to address Smart City scenarios and requirements. There are other works more oriented to the interoperability context that can be used in Smart City scenarios. These works are related with System-of-Systems (SoS) approaches that support different heterogeneous, stand-alone and largely independent constituent systems [24, 25, 26, 27, 28]. While there has been significant research into SoS middleware, these works have not addressed the needs, and complexities of smart cities.

3. CityMii Middleware for Smart Cities

A city becomes truly smart only if it is an autonomous city where the heterogeneous utility systems and sub-systems within the Smart City are able to communicate with each other to operate in an integrated manner. In this context, to achieve a truly Smart City, a common middleware platform is essential.

The CityMii middleware is a centralized hub where the utility systems communicate to their own producers, data collection units, sensors and with each-others (required systems) for synchronized operations. CityMii allows the Smart City authorities to have a centralized control over the data that flows through the various utilities and subsystems. The authorities are able to view the data acquired and handled by these systems. Also, based on defined requirements, the authorities can be able to route the data from a utility system to another system where this data is required.

Since the Smart City systems are foreseen to be deployed for longer periods, sustainability and maintainability are two major concerns. CityMii offers a centrally and scalable management by providing features such as rule based actions, autonomous decision making, real time alerting and status monitoring.

Moreover, CityMii is built over open industry standards, allowing to create a vendor neutral and interoperable eco system.

The overall architecture of the CityMii middleware for Smart City is shown in Fig. 1. It is an integrated architecture of vertical systems for smart cities that allows the visualization of the data in a holistic way, the parameterization of rules and a real-time update in the different vertical systems.

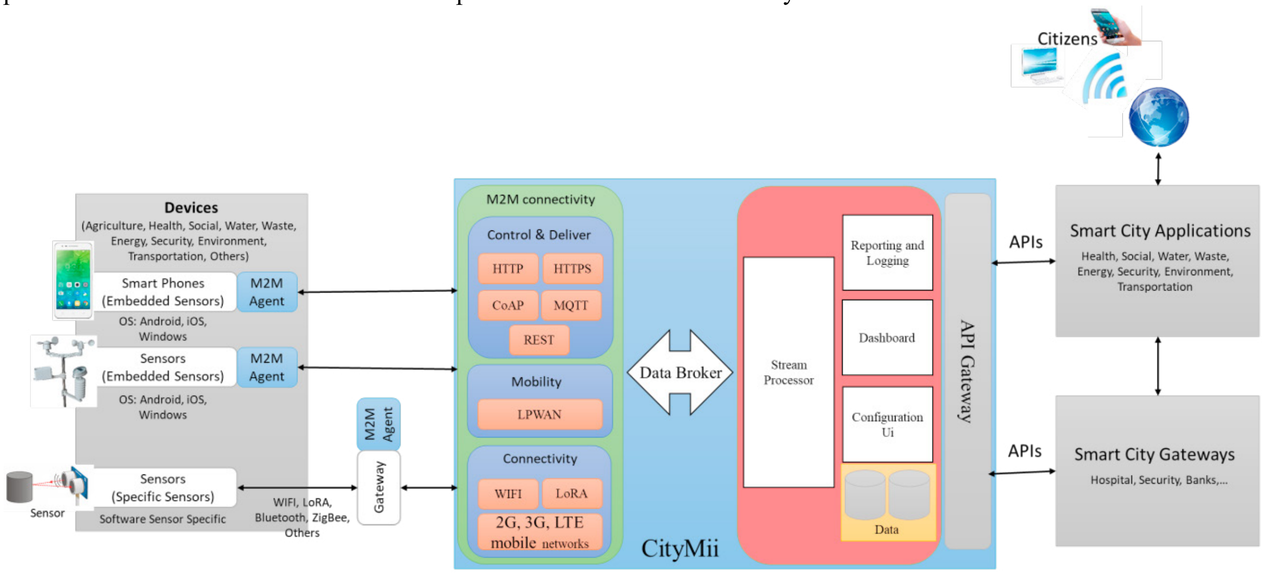

Fig. 1. CityMii middleware architecture for Smart City.

This architecture is composed by three main layers:

- Device layer: that include sensors, actuators and IoT gateways used in different vertical systems in a city;
- Middleware layer: this layer integrates several modules such as the M2M connectivity module, data broker, the configuration UI, reporting and logging, dashboard and the API module.
  - M2M connectivity layer: this module is the entry point to the system. All the data will cross this module to entry or leave the system. It is designed to handle a massive number of concurrent connections and data traffic while the hardware resources are optimized. This module will give back proper acknowledgment to the data senders regarding the status of the data received. The support technologies proposed for the M2M connectivity layer are WIFI, LoRA, 2G, 3G and LTE mobile networks. The module also includes interfaces to multiple types of data points. It has a sensor data unit to receive data from sensors and controllers, an external interface to route data from external systems to the middleware which can be used by any internal subsystem. M2M connectivity module has support for a set of protocols, such as HTTP, HTTPS, REST, MQTT and CoAP. It includes the first level of security including SSL certificate verifications, firewall systems, access control and DoS protections;
  - Data Broker: is a data forwarding system which routes the data to the destination. The destination can be an analysis platform which consumes the data for analysis or other type of application that make use of

data. The broker has the capability to multicast and broadcast data based on the requirement. There is a buffering option for data in case of systems with different data processing speed (e.g. fast data producer and slow data consumer). The data is forward without processing, allowing it to be further processed by other applications.

- Stream Processor: manages data collected by the device layer, process and stores them when needed and sends them to API Gateway to be delivered to the subscribers. This module uses stream structure fields to implement an SQL (structured query language for declarative data management) and ECA (event-condition-action triplets for conditional processing) based approach for selecting which data and which operations to use (computations over data and specifying which conditions should be tested against the data (conditional processing).

- Configuration UI: There is a User Interface (UI) where an administrator is able to easily configure the system. The configuration includes data routing, security settings, processing configurations and alarm/event definition. The UI also shows the system status and its performance.

- Reporting and Logging module: This module handles the reporting and logging of status and errors occurring in the system. Any failure in the system is identified and reported to the administrator. This module is supported by a messaging system that provides email, SMS and mobile app based messaging that make easy to receive system notifications. This module can also be configured using the configuration UI.

- Dashboard: an intuitive dashboard with graphical representations of the subsystems supported by the middleware and its status in real-time. It also represents various parameters of the middleware such as the number of sub-systems currently in the system, number of queues, status of each queue, systems consuming each queue, data flow rate, system utilization and the links to each subsystem UIs & dashboards.

- API Gateway module: this module allows exposing the data from the lower layers to application layer. The API has support for Web Services, REST and HTTP to easily interface with external Smart City client applications.

- Application layer: uses IoT data to create applications with added value for the citizens.

4. Smart City Applications

In order to achieve the real potential of a Smart City, the city must be a platform of applications, incubating ideas of its citizens aiming to promote citizenship. It should stimulate the knowledge of the citizens, enabling its motivation to use and develop new solutions that improve the cities. This is particularly relevant in vital areas such as traffic, tourism, safety, health, environment, public space, culture, education or social activities.

Applications for smart cities are similar to conventional mobile applications. However, according to the current trends they would integrate its data within a common platform, in order to give a global view of the field/sector in a city. Moreover, those applications should be oriented to empower citizens and/or to manage resources within the cities.

The adoption of the CityMii middleware can leverage the development of several new applications able to support city authorities and/or improve the quality of life of people that lives in these smart cities. Following, some examples of application aiming to provide city information to its citizens in order to raise awareness of some of the daily problems found by the city authorities, are described.

Taking into account a “Waste management” scenario, currently, the collection of waste is made by local authorities or contracted companies for this purpose while the consumer is requested to pay a standard price. This price may be the same for all consumers or can be determined by each consumer's location, volume of consumed water or electricity thus not representing the real amount of waste produced by the consumer. One useful application that can be developed in the context of smart cities will be able to quantify the amount of waste produced by a consumer in order to determinate the value that must be paid. In this application, each waste container has an embedded application-activated electronic lock and a weighing-machine, which allows to open the waste content and to measure the amount of waste introduced by each consumer. Defining a garbage separation policy (e.g., common containers are more expensive that separation/specific container), an application that measure the weight introduced in the container and shows the operation cost, allows to make the consumer aware of the importance of separating the waste. This allows to improve the waste management system and the environment in the city, promoting also sustainability.

Other possible application scenario is to develop and deploy environmental monitoring applications providing environmental indicators (e.g., rains and abnormal winds) to the citizens. In addition, for example, the application can also provide and correlate data with the city's green spaces (values/indicators taken from the irrigation system) in order to inform the citizens about the amount of water needed to maintain a given space.

Other scenario is related with the monitoring and control of flow of vehicles and people in city centers, with particular attention on historic centers. In order to do so, it is possible to develop applications that inform the drivers about the quantity of vehicles in a given area, avoiding undesired delays in traffic. Other possible scenarios are to monitor the amount of light available in the city aiming to help people to define better ways, and to avoid assaults, run-ins or other unwanted accidents.

From the above it becomes clear that within a city there are numerous possibilities for the development of innovative smart application. The CityMii middleware architecture also contributes to an effective use of the available multiple data silos existent on the city to effectively manage the city and improve citizen’s awareness.

5. Testing and results

The CityMii middleware architecture was already implemented as a proof of concept and submitted to an initial assessment. The experimental setup scenario defined for this assessment included the data gathered from 10 weather station and 45 mobile phones. Each weather station was installed in a specific location (near green spaces) and collects and transmits a data sample every second. Each sample from the weather station contains the topic id, weather station id, GPS coordinates and the temperature and humidity as measured in the location. The topic id indicates the class of application the data fills.

Three different types of mobile application were also developed for this assessment. The first, application (App1) collects GPS position, accelerometer and gyroscope data and sends it to the CityMii. The second (App2), collects GPS position and the amount of time a user settled in each location. In order to filter possible oscillations in the position it is assumed that if a small variation in the position occurs, the app keeps the location unchanged). The third Application (App3) simulates a common application for social networks by publishing data on the CityMii. It was defined that the mobile applications App1 and App2 collect data and send it every second. In order to make the application heavier (perhaps more realistic), in these two applications we added 500 Bytes of fake data to the sensed data before send it to the CityMii. The third mobile application (App3) allows to send data from mobile phone cameras (images and/or videos) to the CityMii.

The mobile phones were carried by students (they used their own mobile phones) and the data was collected during the class activities. The students were divided into three groups, each group using the same mobile application, App1, App2 and App3.

Three prototypes of smart city applications were developed allowing to display the received data produced by each group of mobile applications and the weather stations. In short, four different data sources (topics) and three consumers (smart city applications) were created according to Table 1. Smart city App 1 and 2 are two conventional applications that deals only with one specific topic. The third smart city App intended to demonstrate the middleware capabilities in spread data to any application, making the system integrated and interoperable.

The CityMii was configured to receive data from devices using HTTP, HTTPS and MQTT protocols. The data broker was configured to forward the data to the corresponding application through the respective API.

The CityMii middleware and the developed smart city applications were running on an i7-3740QM @ 2.70GHz machine with 8 GB RAM and this experiment ran during around 1 hour.

Table 1. Data topics produced and its consumers.

Topic	Data in the topic	Consumer
W1	Station id, GPS position, temperature and humidity	Smart City App 1, Smart City App 3
PH1	Phone id, GPS position, accelerometer and gyroscope info	Smart City App 2
PH2	Phone id, GPS position, time	Smart City App 3
PH3	Phone id, user alias, [photo or video]	Smart City App 3

In terms of results the data from the registered devices were received successfully at the middleware and routed to the corresponding application modules. Fig. 2 and Fig. 3 show the throughput and delay results gathered from our test. The message rate and load is not significant to evaluate the performance of the middleware. However, it intends to show that all information was delivered correctly and there weren't losses in any part of the system. In terms of delay, the system also sustain a good response, introducing a neglected delay (in average, 5.31E-05 ms and maximum 9.22E-04 ms). A large amount of this delay is consumed into each Smart City App and is mainly due to information parsing and processing.

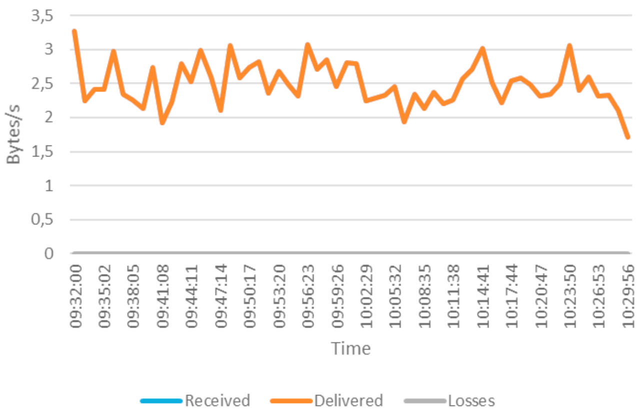

Fig. 2. Message rates (received, delivered, losses).

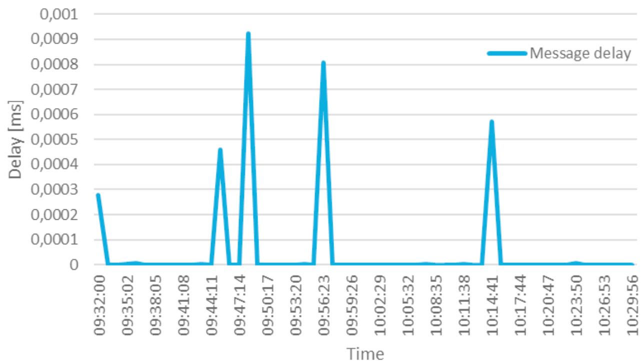

Fig. 3. Message delays.

With this proof of concept, we show that CityMii is able to deal with different classes of applications and it is a viable solution for future cities, providing integration capabilities, which makes the cities smarter and promotes the development of new applications oriented to the real needs of citizens.

6. Next Steps

This work is still under development. Future work includes: (i) designing more drivers to abstract the existing solutions of each specific stakeholder; (ii) optimize each middleware components; and (iii) study other technologies to describe execution flow, to communicate brokers with each other, to describe interfaces between brokers and respective systems, to describe interfaces between brokers and other smart cities platforms' components.

7. Conclusions

Since smart cities use information and communication technology to manage their urban space, a set of sensors, actuators and applications are needed. In order to create an efficient ecosystem, an architecture defined by a set of functional entities that guarantee a holistic, but also detailed view of the city to ensure efficient management by local authorities is also needed. In this context it is expected that systems can work with each other, exchanging data and features. However, it is not easy to create conditions for an efficient collaboration and information exchange among those systems since they are not usually designed to be integrated. The divergence of the adopted technologies and the need for specific knowledge about third-party systems difficult the task of developing interoperable systems. This paper proposed a middleware approach to support the creation, deployment, execution applications for Smart City environments. It supports interoperability among different systems operating in a city.

The model advances the state-of-the-art since it has capabilities to integrate any existent Smart City Application in order to create a single ecosystem able to manage whole city. We have described each module of the system and used an experimental testbed that show the system is able to deal with different class of applications. From our test run, we extracted logs and displayed the incoming and out coming throughput, as well as the delays that proves correct integration of different data sources.

Acknowledgements

This article is a result of the CityAction project CENTRO-01-0247-FEDER-017711, supported by Centro Portugal Regional Operational Programme (CENTRO 2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

References

1. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Gener. Comput. Syst. 2018, 78, 641–658.
2. Rashid, B.; Rehmani, M.H. Applications of wireless sensor networks for urban areas: A survey. J. Netw. Comput. Appl. 2016, 60, 192–219.
3. Saleem, Y.; Crespi, N.; Rehmani, M.H.; Copeland, R.; Hussein, D.; Bertin, E. Exploitation of social IoT for Recommendation Services. In Proceedings of the IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 359–364.
4. Govoni, M.; Michaelis, J.; Morelli, A.; Suri, N.; Tortonesi, M. Enabling Social- and Location-Aware IoT Applications in Smart Cities. In Proceedings of the 2nd EAI International Conference on Smart Objects and Technologies for Social Good, Venice, Italy, 30 November–1 December 2016.
5. Adeleke, J.A.; Moodley, D.; Rens, G.; Adewumi, A.O. Integrating Statistical Machine Learning in a Semantic Sensor Web for Proactive Monitoring and Control. Sensors 2017, 17, 807.
6. Caballero-Águila, R.; Hermoso-Carazo, A.; Linares-Pérez, J. Optimal Fusion Estimation with Multi-Step Random Delays and Losses in Transmission. Sensors 2017, 17, 1151.
7. Guijarro, L.; Pla, V.; Vidal, J.R.; Naldi, M.; Mahmoodi, T. Wireless Sensor Network-Based Service Provisioning by a Brokering Platform. Sensors 2017, 17, 1115.
8. Wu, Z.; Xu, Y.; Yang, Y.; Zhang, C.; Zhu, X.; Ji, Y. Towards a Semantic Web of Things: A Hybrid Semantic Annotation, Extraction, and Reasoning Framework for Cyber-Physical System. Sensors 2017, 17, 403.
9. Choi, H.-S.; Rhee, W.-S. IoT-Based User-Driven Service Modeling Environment for a Smart Space Management System. Sensors 2014, 14, 22039–22064.

10. Floris, A.; Atzori, L. Managing the Quality of Experience in the Multimedia Internet of Things: A Layered-Based Approach. Sensors 2016, 16, 2057.
11. Liu, J.; Hu, Y.; Wu, B.; Wang, Y.; Xie, F. A Hybrid Generalized Hidden Markov Model-Based Condition Monitoring Approach for Rolling Bearings. Sensors 2017, 17, 1143.
12. Guo, Y.; Chen, X.; Wang, S.; Sun, R.; Zhao, Z. Wind Turbine Diagnosis under Variable Speed Conditions Using a Single Sensor Based on the Synchrosqueezing Transform Method. Sensors 2017, 17, 1149.
13. Nie, P.; Dong, T.; He, Y.; Qu, F. Detection of Soil Nitrogen Using Near Infrared Sensors Based on Soil Pretreatment and Algorithms. Sensors 2017, 17, 1102.
14. Bellavista, P.; Giannelli, C.; Zamagna, R. The PeRvasive Environment Sensing and Sharing Solution. Sustainability 2017, 9, 585.
15. Hu, Q.; Wang, S.; Bie, R.; Cheng, X. Social Welfare Control in Mobile Crowdsensing Using Zero-Determinant Strategy. Sensors 2017, 17, 1012.
16. A. Saikar, M. Parulekar, A. Badve, S. Thakkar and A. Deshmukh, "TrafficIntel: Smart traffic management for smart cities," 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI), Pune, 2017, pp. 46-50.
17. Villanueva, F. J., Santofimia, M. J., Villa, D., Barba, J., and López, J. C. 2013. Civitas: The smart city middleware, from sensors to big data. In Proc. of int.. conf. on IMIS, 445-450.
18. Delicato, F., Pires, P., Batista, T., Cavalcante, E., Costa, B. and Barros, T. 2013. Towards an IoT ecosystem. In Proceedings of the International Workshop on SESoS. 25-28.
19. Lea, R. and Blackstock, M. 2014. CityHub: a cloud-based IoT platform for Smart Cities. In International Conf. on Cloud Computing Technology and Science, IEEE. 799-804.
20. Mitton, N., Papavassiliou, S., Puliafito, A. and Trivedi, K. 2012. Combining Cloud and sensors in a smart city environment. EURASIP Journal on Wireless Communications and Networking, 1–20.
21. Petrolo, R., Loscri, V., and Mitton, N. 2014. Towards a Smart City based on Cloud of Things. WiMobCity - International ACM MobiHoc Workshop on Wireless and Mobile Technologies for Smart Cities. Philadelphia, USA.
22. Grace, P., Bromberg, Y. D., Réveillere, L., and Blair, G. 2012. Overstar: An open approach to end-to-end middleware services in systems of systems. In ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing. Berlin, 229-248.
23. Maier, M. 1996. Architecting Principles for SoS. In Sixth annual International Symposium of the International Council on Systems Engineering. Boston, MA, 567–574.
24. Jamshidi, M. 2011. System of Systems Engineering: Innovations for the Twenty- First Century. Vol 58.
25. Blair, G., Bromberg, Y., Coulson, G., Elkhatib, Y., Réveillere, L., Ribeiro, H., Riviére, E., and Taiani, F. 2015. Holons:Towards a Systematic approach to composing SoS.
26. Coulson, G., Blair, G., Elkhatib, Y., and Mauthe, A. 2015. The design of a generalised approach to the programming of SoS. In WoWMoM. 1-6.
27. Curry, E. 2012. System of systems information interoperability using a linked dataspace. In 7th International Conference on System of Systems Engineering.101-106.
28. Billaud, S., Daclin, N., and Chapurlat, V. 2015. Interoperability as a key concept for the control and evolution of the System of Systems (SoS). In International IFIP Working Conf. on Enterprise Interoperability. 53-63.