Semantic Maps for IoT Network Reorganization in face of Sensor Malfunctioning

Fabio Lopes
Department of Electrical Engineering, FCT, NOVA University of Lisbon, 2829-516 - Caparica, Portugal
fas.lopes@campus.fct.unl.pt

Jose Ferreira
Centre of Technology and Systems, CTS, UNINOVA Campus da Caparica, 2829-516 Caparica, Portugal
japf@uninova.pt

Ricardo Jardim-Goncalves
Centre of Technology and Systems, CTS, UNINOVA Campus da Caparica, 2829-516 Caparica, Portugal
rg@uninova.pt

Carlos Agostinho
Centre of Technology and Systems, CTS, UNINOVA Campus da Caparica, 2829-516 Caparica, Portugal
ca@uninova.pt

***Abstract* — As technology evolves, the Internet of Things (IoT) concept is gaining more importance for constituting a foundation to reach better connectivity between people and things. For this to happen, certain strategies and processes are considered to enhance and grant optimal interoperability between the heterogenous devices of a typical IoT network. Two major key aspects of these networks are autonomous error recovery and network reorganization, which are usually based on physical redundancy and aim to return the network to a similar working state, as it was before the error. This process is of great importance when regarding the amount of data and devices that the ever-growing IoT networks have to manage and the number of situations that are associated with this aspect. This work proposes a solution to integrate the previously mentioned processes in IoT networks with the support of the semantic maps as a mean to accomplish redundancy with the use of network metadata and function-oriented recovery methodology, providing the network with tools to be more autonomous and reliable, without compromising performance.**

Keywords — Internet of Things (IoT), IoT Network, Sensors, Semantic Maps, Network Reorganization, Fault Detection

I. INTRODUCTION

The Internet of Things (IoT) is an emerging technological topic that aims to combine consumer products, sensors, industrial components and other everyday objects with Internet connectivity and powerful data analytic capabilities that have the ability to transform the way we work and live [1]. With this, objects become optimized, as every extractable information becomes a mean of analyzing and computing the functioning processes. The results of such analysis are oriented to provide methods for increasing performance, enhanced context functioning and new purposes that come to play when connecting a certain object to an intelligent network.

The previously mentioned topic is an ideal concept, with immeasurable potential, of how people can be connected to objects, that are also connected among themselves, that are best suited to perform under the consumer specifications and the context. Contemporarily and due to different manufacturing specifications and lack of common vocabulary, the heterogeneity of electronical devices is a major issue when designing an interoperable and scalable IoT network. To overcome this and other aspects that arise when applying the IoT concepts to a network, the standardization, provided by well-designed IoT models and ontologies, is a key component to accomplish proper communication. These ontologies suggest configuration methodologies for the network communication processes and should consider a wide variety of devices that communicate using different protocols and technologies, often measuring different things in very different ways and, usually, are not capable of autonomously exchanging information with other devices easily, due to being developed for specific situations. Thus, these models (such as IoT-A, W3C SSN Ontology and IoT Lite [2]) are important guidelines for the creation of IoT networks and should be considered when implementing any kind of methodology that regards the concept of IoT, as the one present in this work.

After the proper configuration processes, the IoT network should maintain an operable working state that corresponds to the high number of devices associated with various IoT applications. The failure of a single sensor should not compromise a whole network and, due the high physical redundancy of having many devices, should be surpassed. To accomplish this, the error must be acknowledged, understood and should result in a unharmful and efficient network reorganization. This work aims to propose a contribution to this process using a concept definition usually regarded in areas of linguistics and learning, the semantic maps, which intend to provide meaning and knowledge to words in the same semantic field. In technological terms, the construction of semantic maps can be a contribution to achieve an intelligent and efficient network monitoring system, especially in challenging contexts of big data, where there are many heterogenous sources that produce too much information to be manually reviewed [3]. The idea behind the use of this concept is to gather and specifically organize meta-information, regarding the network and its components, to allow redundancy by recovering from device failures. This recovery is possible by using the semantic maps as a source of information to this process.

II. BACKGROUND

To specify and better understand the methodology proposed in this paper, some key concepts must be mentioned and explained. The objective of understanding and explaining these concepts is to understand how they can contribute to the creation of semantic maps. These subjects are related to the aforementioned environment of IoT and, mainly, denote methodologies and concept processes to accomplish what is suggested in this work.

A. Knowledge Mapping

Knowledge mapping is a concept of organization and categorizing the existing information in more usable and accessible formats [4]. The idea of its use in networks, is to create a dynamic structure that accompanies the functioning of the network, elucidating and assisting with the analysis of the ongoing complex processes [5]. This synthesis of knowledge should be both essential and critical to the processes, using the information value as a criterion to the mapping process.

B. Fault Detection

Fault detection is the first process when tackling the autonomous error recovery of a network component. The failures, or errors, in IoT networks can be very different and cannot be completely generalized. The detection of sensor failures is usually possible by autonomous learning of patterns, or pre-configured settings, and is carried out by checking expected thresholds and consistency among the similar and redundant sensor measurements [6]. The information that is analyzed for the fault detection is, typically, the data streams and events generated by measuring devices, such as sensors.

C. Systems Self-Organization

After isolating the detected anomaly, the process of network reorganization begins. Systems self-organization is a prominent technological concept that is defined by the autonomous processes of decision-making to adapt to context and aspects of the system deployment environment [7]. The characteristics and processes that this concept may include exceed the needed information to understand the methodology suggested in this paper. The important aspect to regard is that it aims to perform processes to maintain the correct operation of a system and more specifically to the context of this work, it can be used to autonomously restore the functioning of a network after a sensor malfunctioning, being able to recover the harmonization of the IoT network after a fault detection.

Every time that there is a sensor fault detection, some changes occur, during the operating state of the network, thus, the terms context awareness, self-configuration and self-adaptation are important to consider. Context awareness refers to the property of a device to passively or actively determine its context, which may not be clearly defined. Self-configuration is based on initial sensor configurations regarding the user specifications and includes methods for generating dynamic configurations that better suit the ongoing situation [8]. And lastly, self-adaptation is bind to the term self-management and refers to the ability of an entity to calibrate its functioning to correspond to the environment where it was deployed [9].

D. Complex Event Processing (CEP)

The CEP is defined by three steps: registration of event sources, the definition of EPA's (Event Processing Agents) and the registration of event sinks. The event sources are responsible for providing the data from the monitored environment, generating events. The EPA's, the core of event processing, process the input from the event sources and try to detect situations of interest (SOI), set by pre-defined rules, integrated in the CEP database, regarding the context of the implementation. It is important to mention that the CEP mainly focus on detecting SOI in streaming data rather than manipulating data streams [10] and static CEP, contemporarily predominant, are very context-sensitive [11]. Thus, ordinary context-based events will be discarded and the events that express importance, typically considerably less, will be submitted to analysis regarding persistent queries directed to the, already mentioned, rules. The results of such analysis, originates actions that take place in the event sinks, to whom the rules are specifically oriented to. An event can be defined as a significant change of state [12] and the use of a CEP and focus on device originated events, makes the suggested architecture in this work an Event-Driven Architecture. In this work, CEP is used to detect specific sensor key performance indicators, to support the system in the sensor fault detection and enabling the reorganization of the IoT network.

III. SEMANTIC MAPS FOR IOT NETWORK REORGANIZATION

As mentioned previously, semantic maps are regarded as a strategy to represent concepts [13]. The process of creating such maps is called semantic mapping. In technology, since the internal representation of information gathered by technological components is not intuitively understandable by humans and vice-versa, and it is inadequate for learning process (in its raw form), the combination of object classification and common-sense knowledge makes the semantic maps an interesting approach to create a useful network description. Thus, the information included in the configuration and representation of meta-information of the network components in semantic maps provides expressive information, contextual integration of the observations and correlation of the knowledge of the environment [14].

A. Event Processing

Generally, systems that rely on processing the information of deployed devices and their analysis, to read or predict conditions that could trigger pre-determined rules, recur to the integration of a CEP module. An CEP, module analyses large flows of primitive events received from a monitored environment to timely detect situations of interest [15]. This processing takes place following user-defined rules and that aspect induces a liability to the dynamism of the general system, because any change may cause an incompatibility to adapt or, in the case of a device failure, the rules may become obsolete due to a non-implementation of redundancy, at same time can be used to detect possible problems. In some situations, these obsolete rules can be view as a problem in the sensor, giving the possibility to be used to detect failures and warn the users about it.

Generally, the CEP module listens to incoming events generated by the devices and, following the pre-determined rules (generated by the developer), detects situations that may trigger events on the devices or simply alert the system monitor. The idea behind understanding the CEP is to perceive its importance in the application of the IoT models, and to provide the intended methodology that this work aims to propose, the use of semantic maps. These semantic maps, as mentioned, are intended to improve and test the trustworthiness of the system by standing as a tool to overcome errors and failures in devices or, more specifically, in sensors. The idea for the functioning method of these maps is to be a representation of similar configurations for the network (using other available sensors or alternative configurations).

B. Sensor Malfunction

Recurring to a brief example, let's say sensor B, depicted in Fig. 1, is damaged by unknown reasons and provides unusable information (e.g., temperature readings out of predictable thresholds or an anomaly detected by comparison with nearby devices). Sensor B is now a liability to the event processing and may cause several rules to become obsolete for not having the necessary information to be triggered. To tackle this situation, human intervention is usually needed. In large industrial networks or in the IoT paradigm, there are numerous sensors, deployed for different uses, that may be suited to comply with operations related to Sensor B. To measure that possibility, sensors can be analyzed regarding aspects like localization, type of measurement, role in the network and others. In this case, Sensor A and C, serve the same purpose and are suitable to replace sensor B in rules associated with it, as well as other similar sensors that measure temperature.

This (Fig. 1) is a simple situation, manageable by human intervention if it occurs sporadically. The idea behind this work is to present a mechanism that provides autonomous dynamic adaptation. This is where semantic mapping comes to play, because it represents and maps possible redundancies white providing room for developing the autonomous creation of possible maps regarding the aspects of the role of each sensor in the network. This autonomous creation is possible, as we'll be more clear later in this paper, but it's advisable to create the initial configuration of the device database and relevant possible core semantic maps by the developer, to provide consistency to the network.

C. Semantic Mapping Equation

The Semantic MAPping module (SMAP) specifies the semantic mapping and runtime processes, it is important to understand what specifications the semantic maps need to distinguish in the sensors. These specifications are considered by relevant fields summarized in (1).

$$Map = \langle ID(O, D[i]), Mismatch, Role, Correlation, Weight \rangle \quad (1)$$

The following aspects were considered to form the mapping equation:

- *Identifiers (ID's)*, which serve the purpose of the structural organization of the network, sensors and maps involved. It represents the individual unique attribute that differentiates each instance. The letter "O" defines the origin sensor and the letter "D" defines the destination sensor(s).
- *Type of relation or association (i.e. mismatch)* is a representation of the difference between each instance in a way that highlights the aspects that may have to be considered while mapping.
- The *role of the original instance* is very important, perhaps the main factor to be considered, because it represents each functionality that the origin sensor has in the network, i.e. every function in the network that needs to be mapped (one map for each). After the map is used, the destination sensor(s) add the new role to their device information.
- *Output data differences between sensors (correlation)*, is the relation between the output that each sensor may have, regarding its specifications (e.g. voltage output) and information to be considered when analyzing the data (e.g. average between measures).
- The *importance weights in decision-making* is a way to differentiate several maps for the same role of the same sensor. This value is dynamically adjusted during the operation of the system (e.g. a sensor with too many roles assigned gets its value decremented to avoid too much reliance by the network) and the map with higher value is chosen.

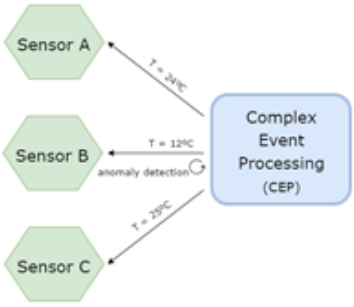

Fig. 1. *Sensor Malfunction Example.*

IV. SMAP Architecture

In this section, a general system architecture is presented (Fig. 2). For this design, all the mentioned functionalities and processes were considered in addition to how contemporary systems of this kind are defined. It is important to mentioned that the fact that the SMAP Module, explained further in this section, is defined separately from the rest of the system and somehow as autonomous as possible, making it not mandatory for a system to implement the SMAP module at an early stage, being possible to implement it on existing systems.

A. CEP Engine

The CEP module was mentioned earlier in this paper and to avoid repeating the basic and generic notions of the concept, the approach here will be simple and from a functional point of view. It is also important to mention that some event

processing, regarding the verification of correct sensor operation, can be done by the semantic mapping module, allocating some working load from the CEP.

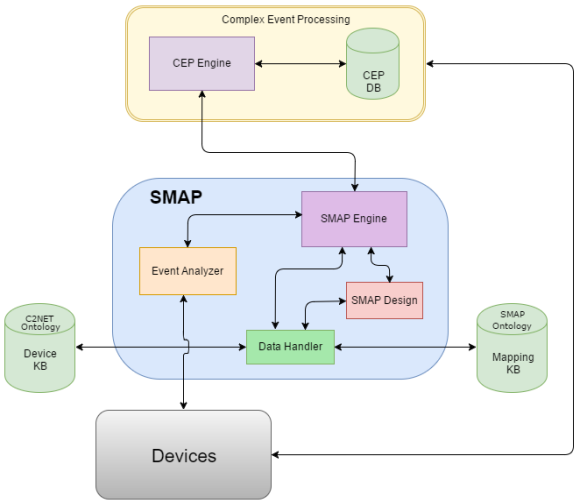

Fig. 2. *System Architecture*

B. Devices

The devices module is the low-level architecture that represents the technology that provides and generates events to be considered by the CEP and SMAP modules. These events are usually raw information about the monitored environment and make no judgement or evaluation about it. This module is also connected to the semantic mapping module because of the need to keep an up-to-date device database that may also be updated, autonomously or manually, according to specific changes that may occur in the devices.

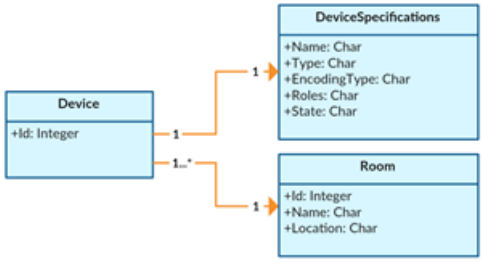

Fig. 3. *Device Knowledge Base Class Diagram.*

C. Device Knowledge Base

The main objective of this KB is to keep an updated registry of the sensor's information (e.g. the current roles). In Fig. 3 is represented a class diagram that considers the essential specifications for the operation of the system. The idea behind having a specific KB for this information is to specify and contextualize the sensors dynamically, during the functioning of the system and use of semantic maps. This KB follows specifications developed by the C2NET project [16].

D. Mapping Knowledge Base

Similarly, to the previous knowledge base, the objective of this KB is to keep updated information about the current mappings. It stands as a dynamic record, updated every time a change is made to the network, and is not particularly relevant for any other decision-making processes. In Fig. 4, the structure of this KB is better detailed. The connection between sensors and mapping is not as direct as may seem and is provided by the semantic mapping module. Nevertheless, it is explicit that for each mapping there is only one origin sensor and may be one or more destination sensors.

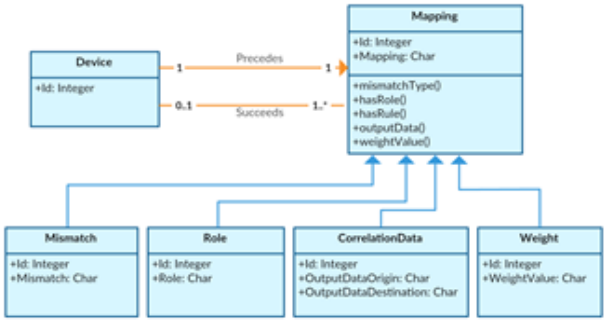

Fig. 4. *Mapping Knowledge Base Class Diagram*

E. Semantic Mapping Module - SMAP

The SMAP module stands as the module that represents the concept and methodology of this work. It interacts with the devices, mapping knowledge base, device database and the CEP module, specifically the CEP engine. The main functions of this module are presented in this section:

- *Specific event listening from CEP and Devices* - These events are analyzed according to situations of interest regarding the monitored environment, the potential of the developed semantic maps and the pre-determined mapping situations.
- *Verification of sensors* - The SMAP module, like the CEP, will process events that interest to the objective of its existence. The main interest is to analyze the state of the sensor, to check if it is functioning properly. To accomplish this, it occurs a loop of event listening and processing, until some situation triggers a semantic mapping procedure. Some pre-defined failure situations can be, e.g., measurements that exceed typical values or thresholds, constant measurements, noisy readings or non-concordant measurements between two or more sensors of the same type, in the same area [17][18].
- *Search and update queries to device database about origin sensor* - In order to keep an updated record of the sensors and to search the correct mappings, the semantic mapping module searches the available information about the sensor that expressed a failure, including the roles that is operating at the moment, and updates them to "none" and changes the state of operation of the device to Boolean zero.
- *Search and update queries to mapping knowledge base* - After acquiring the information about the origin sensor, the SMAP module elaborates the request for a semantic mapping solution. To accomplish this, it searches the role, or roles, of the origin sensor in the mapping knowledge base. If there are no semantic maps

for that type of role, manual intervention is solicited or mechanisms of autonomous mapping are triggered. After finding the semantic maps for the needed role, the maps that have the mention sensor as the origin sensor are selected, by adding constraints to the previously mentioned query. The remaining maps, if more than one, have a weight component associated to them, as mentioned before, and the map that has a higher weight value is selected for the semantic mapping process.

- *Search and update queries to device database about destination sensor(s)* - Similarly to happens when using the device database for the origin sensor, the semantic mapping module queries the device database for information about the destination sensor: It retrieves the current state to verify if it is indeed available (normally it is operating because the device database should have updated information. The roles may also be considered to avoid too much reliability on a solution, but again this is controlled by the weight value in sensor database and it is typically updated regarding that issue. If everything is according to the specifications for good functioning, the sensor is updated within the database with a new role and state, if that is the case.
- *Sensor output correlation* - In this phase, within the semantic mapping module, the correlation between outputs from the origin and destination sensor is considered recurring to the output data and mismatch from the mapping information, retrieved from the mapping knowledge base. Any particular change or specification that the CEP has to deal with, in the event processing or event rules that use the destination sensor considered, are taken into account.
- *Event update in CEP* - In this final phase, the information about the destination sensor, or sensors, is provided to the CEP to make the necessary changes to the events, previously using the origin sensor and replacing it. The information considered in the last point, sensor output correlation, is also provided to the CEP in order to integrate them in the mentioned events.

Considering the mentioned points, it is logical to define sub-modules according to the specifications and functions of the SMAP, as illustrated in Fig. 2. The first two points, regarding the analysis of SOI, are responsibility of the "Event Analyzer" module. Every change or query made to the device KB and mapping KB use the module named "Data Handler". The core runtime process and, generally, all other module processes, are conducted by the "SMAP Engine". In addition, the module "SMAP Design" is oriented to the design time of semantic maps by means of an interface or autonomous processes by an human user.

V. Implementation of the Use Case Scenario

The application of the designed module was envisioned while developing it. It assumes the functioning described earlier but it does not demand a strict implementation and is adaptable to different situations and contexts, within technological environments and the IoT paradigm. Its features were designed in order to help the network, where it is implemented, to have scalability and handle the constant growth and diversity of IoT, without being too susceptible to failures.

To demonstrate this scenario of a smart room of a modern factory that has various machines and sensing devices that allow to control its functioning, and to provide better working conditions for the employees. This type of "aware" room is part of what is now called the Industry 4.0 (or fourth industrial revolution) and it is integrated in the IoT paradigm, aiming to connect embedded system production technologies and smart production processes with increased connectivity and ever more sophisticated data-gathering and analytic capabilities [19]. Table 1 is provided some insight of sensors and objectives, that the sensing devices, along with technological modules (like the one presented in this paper), may improve and offer to this type of environment. The semantic mapping module, specifically, is intended to allow reliability to these processes, managed according to the rules specified by the developers in the network, with redundancy by using different devices, already deployed, to assure the continuous manufacturing process and wellbeing of the employees.

Table 1 Possible objectives in the presented scenario

Objective	***Devices***	***Example of Rule and Action***
Adequate Room Temperature	Temperature Sensor	Room temperature is high, the air conditioning is activated.
Fire Detection	Temperature Sensor and/or Gas Sensors	Temperature and/or smoke density indicates a fire, fire alarm is activated
Room Light Adjustment	Photoresistor or Photo Diode Light Sensors	Light is adjusted according to a pre-defined value.
Security	Ultrasonic Sensor or Passive Infrared Sensor	Safety Alarm is activated when an intruder or non-authorized person is detected.
Gas Leakage or Oxygen Depletion	Specific Gas Sensors (like CO2, CO, O)	The concentration of certain gases is life-threatening, the ventilation is activated.
Detect Safety Distance to Machinery	Ultrasonic Sensor or Passive Infrared Sensor	Distance to machine is not satisfied, alarm is activated and machine stopped.
Tracking Products, Tools or Persons	RFID, Nano Tags or Image Capture and Recognition	Determining current states of the production process and person safety.

As Table 1 shows, there are multiple and possibly infinite applications of this processes and methodologies. The objectives column, regarding the implementation of the semantic mapping module, is directly related to the role field in (1). Thus, the role is a designation of this objective and the right column sheds some light of a possible association of that role to a pre-determined event rule. The center column shows a suggested use of devices, used singularly or together, to allow measuring and determine the values that will be compared to the patterns delimited by the mentioned rules.

To exemplify the process, recurring to an ordinary example based on Table 1 combining temperature and gas measurements for fire detection, Fig. 5 shows a simplified view of the process. The CEP of a certain network is getting measurements of Temperature Sensor A, CO2 Gas Sensor A and CO Gas Sensor A to get a viable analysis to whether there is occurring a fire, on a certain factory machine zone, or not. In

this situation, Temperature Sensor A is measuring very high values while the gas sensors do not detect any anomaly. The SMAP module detects this SOI and compares the reading with a nearby similar temperature sensor, deployed for a different reason. This comparison shows that Temperature Sensor A is not functioning properly and the output is a suggestion of the sensor's malfunctioning. After this, the SMAP uses the proper processes to query the knowledge bases of sensors and semantic maps to determine a viable solution to maintain the efficiency level of this fire detection and other roles in which the Temperature Sensor A is used for. The Temperature Sensor B, used for verification, is mapped as a redundancy and the SMAP knowledge bases and the CEP's rules database are then updated to replace Sensor Temperature A with it. Finally, the CEP maintains the previous operation of fire detection, using Temperature Sensor B instead of Temperature Sensor A.

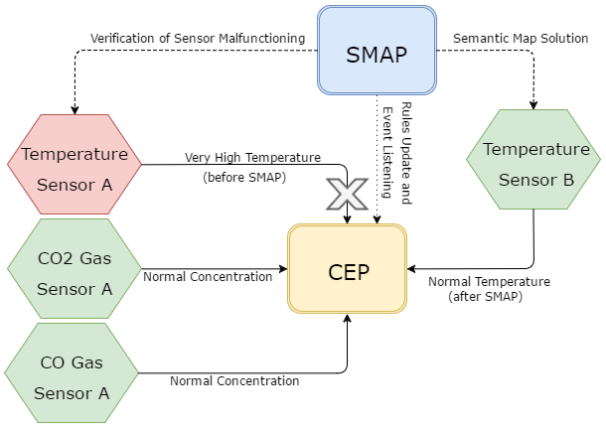

Fig. 5. *Scenario Example*

VI. Conclusion and Future Work

In this work, a suggested methodology for using semantic maps, recurring to network metadata and function-oriented recovery methodology, to accomplish autonomous error recovery and network reorganization, is presented. The objective of this process is to improve reliability and trustworthiness regarding IoT network monitoring systems, without compromising performance or significant structural changes to already implemented networks. These aspects were considered during the design of the SMAP module and its interactions with the rest of the system. To contextualize and demonstrate the developed work, a mapping, and the use of the resulting map, was described to represent the functionality and to show the viability of use of this solution. This implementation, regarding the validation of the objectives of this work, shows the capability of the network, when using the semantic mapping methodology, to detect common errors, trigger an error recovery process, analyzing the redundancies provided by the existing semantic maps, and to reorganize the network, to return it to a similar working state, as it was before the error occurred. Another key feature, is the aspect to also leave room to autonomous recognition of error patterns and the autonomous creation of new semantic maps, within the SMAP Design module.

In future developments of this work, it would be interesting to improve fault detection of sensor malfunctions, a process that by itself can lead to a significant degree of complexity, enhancing the possible uses of the SMAP module regarding the detection of SOI and providing more stability to the typical network. Another aspect that deserves attention is the improvement of the process that lead to the autonomous creation of semantic maps, creating possible redundancies with different types of sensors and measurements, to provide a better response to non-typical sensor malfunctions.

Acknowledgment

The research leading to these results has received funding from the EC HORIZON2020 Program under grant agreement n° C2NET 636909 (http://www.c2net-project.eu/) and nº VF-OS 723710 (http://vf-os.eu/).

References

[1] C. L. Rose Jaren, Eldridge Scott, "The internet of things: an overview - Understanding the issues and challenges of a more connected world," *Internet Soc.*, no. October, 2015.

[2] J. Ferreira, J. N. Soares, R. Jardim-Goncalves, and C. Agostinho, "Management of IoT Devices in a Physical Network," *CSCS21: The 21th International Conference on Control Systems and Computer Science*, Bucharest, Romania, 2017.

[3] C. A. Knoblock and P. Szekely, "Semantics for big data integration and analysis," *2013 AAAI Fall Symp.*, vol. FS-13-04, 2013.

[4] V. Powers, "Knowledge Mapping Guides Organizations to Knowledge Within its," *Am. Product. Qual. Cent.*,V2,pp. 2–4, 2002.

[5] S. Ebener and A. Khan, "Knowledge mapping as a technique to support knowledge translation," *Bull. World Heal. Organ. 84(8), pp.636-642.*, vol. 29736, no. 6, 2006.

[6] L. Jiang, M. Liu, and E. A. Latronico, "Sensor Fault Detection and Isolation Using System Dynamics Identification Techniques," 2011.

[7] V. I. Yukalov and D. Sornette, "Self-organization in complex systems as decision making," *Adv Complex Syst*, vol. 17, 2014.

[8] L. Guardalben, L. J. G. Villalba, F. Buiati, J. B. M. Sobral, and E. Camponogara, "Self-configuration and self-optimization process in heterogeneous wireless networks," *Sensors*, vol. 11, no. 1, 2011.

[9] G. Weiss, M. Zeller, and D. Eilers, "Towards Automotive Embedded Systems with Self-X Properties," 2010.

[10] L. Woods, J. Teubner, and G. Alonso, "Complex Event Detection at wire speed with FPGA's." 2010.

[11] B. Hoßbach and B. Seeger, "Anomaly management using complex event processing: extending data base technology," *Proc. 16th Int. Conf. Extending Database Technol.*, 2013.

[12] S. Chakraborty and J. Eberspacher, *Advances in Real-Time Systems*. Springer, 2012.

[13] NSW Centre for Effective Reading, "Vocabulary - Semantic Mapping," 2012.

[14] W3C, "SSN Applications," 2012. [Online]. Available: https://www.w3.org/community/ssn-cg/wiki/SSN_Applications. [Accessed: 01-Sep-2016].

[15] A. Margara, G. Cugola, and G. Tamburrelli. "Learning from the past: automated rule generation for complex event processing" *Proc 8th ACM Int Conf Distrib Event-Based Syst-DEBS14*, 47–58, 2014.

[16] C2NET, "C2NET," 2015. [Online]. Available: http://c2net-project.eu/. [Accessed: 24-Aug-2016].

[17] S. Munir and J. A. Stankovic, "FailureSense: Detecting sensor failure using electrical appliances in the home," *Proc. - 11th IEEE Int. Conf. Mob. Ad Hoc Sens. Syst. MASS 2014*, pp. 73–81, 2015.

[18] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor Faults : Detection Methods and Prevalence in Real-World Datasets," *ACM Trans. Sens. Networks*, vol. 6, no. 3, 2010.

[19] B. Sniderman, M. Monika, and M. J. Cotteleer, "Industry 4.0 and manufacturing ecosystems," *Deloitte Univ. Press*, 2016.