BRAIN-IoT: Paving the Way for Next-Generation Internet of Things

Enrico Ferrera[1], Xu Tao[1], Davide Conzon[1], Victor Sonora Pombo[2], Miquel Cantero[3], Tim Ward[4], Ilaria Bosi[1] and Mirko Sandretto[1]

[1]*LINKS Foundation – Leading Innovation & Knowledge for Society, Turin, Italy*
[2]*Improving Metrics, A Coruña, Spain*
[3]*Robotnik Automation S.L.L., Valencia, Spain*
[4]*Paremus Ltd, Wokingham, U.K.*

Keywords: Next-Generation Internet of Things, Self-aware Systems, Semi-autonomous Systems, Security, Privacy, Edge Computing, Model-based Development, Smart Behaviours.

Abstract: Nowadays, the adoption of the Internet of Things is drastically increasing in different domains and is contributing to the fast digitalization of several different critical sectors. In the near future, next generation of IoT-based systems will become more complex to be designed and managed. An opportunity for the development of flexible smart IoT-based systems that drive the business decision-making is to take more precise and accurate decisions at the right time, collecting real-time IoT generated data. This involves a set of challenges including the complexity of IoT-based systems and the management of large-scale systems scalability. With respect to these challenges, we propose to automate the management of IoT-based systems mainly based on an autonomic computing approach; these systems should implement cognitive capabilities that allow them learning and generating decisions at the right time. Consequently, we propose a model-driven methodology for designing smart IoT-based systems. With this objective, BRAIN-IoT paves the way to develop and demonstrate novel IoT concepts and solutions to underpin the Next Generation Internet of Things vision and architecture, that focusing on self-aware and semi-autonomous IoT systems, as well as on moving away from centralized cloud-computing solutions towards distributed intelligent edge computing systems.

1 INTRODUCTION

Nowadays, the adoption of the Internet of Things (IoT) is drastically increasing in every application domain, contributing to the fast digitalization of contemporary society. Current IoT scenarios are demonstrating to be constantly increasing in terms of demanding non-functional requirements, from low latency to high reliability, and dynamic resources allocation. This paradigm shift, also considered as the next evolution phase of IoT (Fettweis, 2014), is expected to create numerous opportunities for technology market supporting applications such as critical infrastructures management, and cooperative service robotics.

To cope with these demanding requirements, a multitude of novel technologies - such as Edge Computing, Artificial Intelligence and Analytics, Digital Twin, as well as Security, Privacy and Trust schemes – are being investigated in order to be adopted in current IoT architectures standards (Vermesan & Bacquet, 2019), identifying efficient integration schemes with proper design patterns. Henceforth, next generation of IoT-based systems are set to become more complex to design and manage.

Nonetheless, a set of challenges including the complexity of IoT systems in domains like Smart Factories and Smart Cities and the management of the possibly conflicting requirements, interoperability among distributed heterogeneous technologies and data, as well as system scalability, requires an evolving software eco-system which can adapt, change and scale in response to local requirements and environmental changes to drive efficiently the business decision-making. Also, the distributed nature of IoT makes enforcement of good security practices intrinsically challenging. The market asks for IoT solutions suitable to safely support business-critical tasks, which can be deployed rapidly and with low costs. Modern IoT applications operating in

Ferrera, E., Tao, X., Conzon, D., Pombo, V., Cantero, M., Ward, T., Bosi, I. and Sandretto, M.
BRAIN-IoT: Paving the Way for Next-Generation Internet of Things.
DOI: 10.5220/0009818504700477
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 470-477
ISBN: 978-989-758-426-8

different scenarios, such as Smart Cities, Industry 4.0, etc., are complex software ecosystems with strict requirements of geographic distribution, scalability, heterogeneity, dynamic evolution, security and privacy protection, highly more challenging than the ones required by the traditional (e.g. domotics) environments. Two of the main challenges arising in the current Internet of Things scenarios, on one side, the requirement of designing applications involving several heterogeneous platforms and smart Things interconnected to each other in the same environment and, on the other side, the need to be able to instantiate, operate and evolve the complex software ecosystem, reacting automatically and at runtime to environmental changes, without the human intervention. With respect to these challenges, which relates to the complexity of IoT systems management, the authors propose to automate the management of IoT systems based on an autonomic computing approach. However, autonomic computing alone is not enough for the development of smart IoT-based systems. Indeed, these systems should implement cognitive capabilities that allow them learning and generating decisions at the right time (Kephart & Chess, 2003). Consequently, we propose a model-driven methodology for designing smart IoT systems. With this objective, the BRAIN-IoT project (Brain-IoT, 2018), funded by the European Commission, paves the way to develop and demonstrate novel IoT concepts and solutions to underpin the *Next Generation Internet of Things (NGIoT)* vision, which requires the definition of next generation IoT architectures focusing on self-aware and semi-autonomous IoT systems, as well as the migration from centralized cloud-computing solutions towards distributed intelligent edge computing systems (Ferrera, et al., 2018).

While EU-based initiatives are devoting significant amount of effort to tackle such issues, often with very positive results, efficient solutions suitable to tackle challenges arising in NGIoT scenarios are still missing (NGIoT, 2019). Future critical issues may be hiding under the hood already now and be ready to appear in the close future. To be economically sustainable and achieve solution longevity, a flexible and dynamic self-managed system is needed.

The rest of the paper is organized as follows: Section 2 provides an overview of the background on autonomic computing and current state of the art of the application of MAPE loop approach in IoT domain. Section 3 introduces the BRAIN-IoT Platform architecture starting from a general, functional, overview and continuing with objectives to be accomplished and design choices. Section 4 presents the Service Robotics use-case scenario where the Platform has been applied and tested. Finally, Section 5 and Section 6 draw conclusions and discuss future works to be carried in the next phase of the project.

2 BACKGROUND AND RELATED WORKS

One of the most important challenges in self-adaptation is to create the ability in the system to reason about itself and its context (Abeywickrama & Zambonelli, 2012) (Renart, Balouek-Thomert, & Parashar, 2017).

An autonomic system is composed of many self-managed components that interact with each other autonomously, giving some kind of decision-making mechanism, such as policies, from administrators (Bueno, 2012). The self-management system must be continuously monitoring itself to be aware of changes in the system that might require either reconfiguration or optimization of the components, protecting itself against suspected wrong behaviour or recovering from failures (Villegas & Müller, 2011): BRAIN-IoT focuses on these last two aspects. Adaptation refers to the capability of changing the software structure or behaviour according to significant alterations in the environment. This adaptation must occur dynamically and at runtime.

The MAPE loop (Monitor, Analyzer, Planner, and Executer) (Lee, Seo, & Kim, 2019) is a crucial feedback for the implementation loop of self-adaptive software and autonomic computing. These four parts communicate and collaborate to each other and exchange appropriate knowledge and information. The Monitor collects the information from the software and its environment, the Analyzer observes the reported situation and determines if any change needs to be made, the Planner creates the plan to achieve the changes, and the Executer provides the mechanism to perform the changes over the managed system. Various decentralized MAPE loop patterns have been studied for self-adaptive software such as coordinated control, information sharing, master–slave, regional planning, and hierarchical control.

The following paragraph will report the various IoT studies conducted on self-adaptive software, with a focus on its application in distributed systems and IoT domain. One of the first approaches related to a self-adaptive distributed decision support model is proposed by Zhang (Zhang, Alharbe, & Atkins, 2016) with a simulated inventory management system for a warehousing company. Ouechtati (Ouechtati, Azzouna, & Said, 2018) presented a middleware for IoT, which can adapt process of access control rules that satisfy the requirements of IoT environments. A designed MAPE loop-based management architecture patterns for an adaptation system in IoT environments was designed by Ribeiro (Ribeiro, de Almeida, Moreno, & Montesco, 2016). Muccini (Muccini, Spalazzese, Moghaddam, & Sharaf, 2018) surveyed IoT distribution patterns and self-adaptation, and simulated with an IoT-based forest monitoring system based on the MAPE loop. In the health area (Azimi, et al., 2017) presented a MAPE loop with shared knowledge (MAPE-K) based on hierarchical computing architecture (HiCH) for IoT-based health monitoring systems. Welsh (Welsh, Bencomo, Sawyer, & Whittle, 2014) implemented a self-adaptive system with goal-based requirement models to ensure self-explanation behaviours at runtime. In order to simplify the engineering and coordination of services in dynamic IoT environments in (Beal, Pianini, & Viroli, 2015) the aggregate programming (focuses on ensuring the simplified design, creation, and maintenance of IoT systems) was employed and then demonstrated in the Alchemist simulation (Pianini, Montagna, & Viroli, 2013), which is an extensible meta simulator for pervasive computing. In (Bucchiarone, Marconi, Pistore, & Raik, 2017) was promoted a framework with runtime service composition in a dynamic context, using a service model with stateful, non-deterministic, and asynchronous features. Renart (Renart, Diaz-Montes, & Parashar, 2017) proposed a framework to support dynamic data driven IoT applications called Pulsar, that leverages edge resources to support location and content aware processing of data streams. Also, a self-adaptive framework for reliable multiple autonomic loops was proposed by (Sylla, Louvel, Rutten, & Delaval, 2017).

Furthermore, a new challenge related on self-adaptive software concerns the complexity that arises from the wealth of information that can be associated with runtime phenomena. Trying to extend the applicability of models produced in Model-Driven Engineering (MDE) approaches to the runtime environment, the research on models@runtime focuses on providing effective technologies for managing the complexity of evolving software behaviour while it is executing (Bencomo, Götz, & Song, 2019). A models@runtime is a causally connected self-representation of the associated system, so it is possible to use these to support dynamic state monitoring and control of systems during execution (runtime behaviour observation), to support design errors fixing or even controlled ongoing design. In other words, in order to monitor and validate the correct execution of the platform, models@runtime could provide a real-time picture of the status of the system, synchronizing its internal status with its model (Blair, Bencomo, & France, 2009). This guarantees the system operators to constantly have a clear view of the situation, checking whether the system is behaving as expected, and giving the possibility to promptly react in case of design issues.

3 BRAIN-IoT PLATFORM

3.1 Overview

In this paper, the authors focus on IoT environments consisting of different sensors, actuators, external services, and requirements that should be dynamically satisfied at runtime. To achieve this, a self-adaptive software framework is proposed for an IoT environment with two phases: Modelling & Validation, and Execution. In Figure 1, presents an overview of the proposed platform.

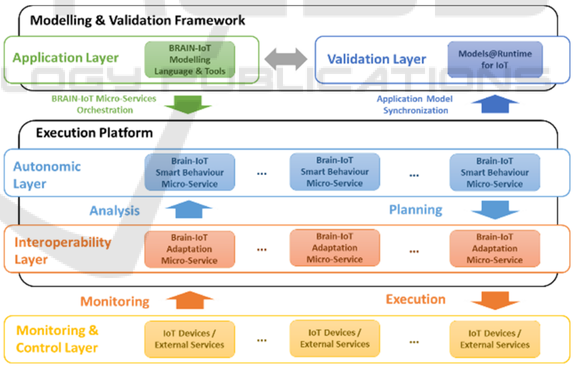
Modelling & Validation Framework
Application Layer
BRAIN-IoT Modelling Language & Tools
Validation Layer
Models@Runtime for IoT
BRAIN-IoT Micro-Services Orchestration
Application Model Synchronization
Execution Platform
Autonomic Layer
Brain-IoT Smart Behaviour Micro-Service
...
Brain-IoT Smart Behaviour Micro-Service
...
Brain-IoT Smart Behaviour Micro-Service
Analysis
Planning
Interoperability Layer
Brain-IoT Adaptation Micro-Service
...
Brain-IoT Adaptation Micro-Service
...
Brain-IoT Adaptation Micro-Service
Monitoring
Execution
Monitoring & Control Layer
IoT Devices / External Services
...
IoT Devices / External Services
...
IoT Devices / External Services

Figure 1: Overview of BRAIN-IoT Platform.

The Modelling & Validation Framework is responsible for designing the logic of the Application which is going to be constructed based on the actions and relations between the available devices (i.e. sensors, actuators, Cyber-Physical Systems - CPSs) and external services (e.g. weather forecast, open data, third-party IoT platforms, databases). The Application logic is modelled along with the relevant IoT environment as a Finite-State Machine (FSM) describing its self-adaptive behaviour. Finally, the abstracted FSM is converted in source code which is deployed and executed by the Execution Platform.

The Execution Platform implements a MAPE loop, which make the Platform dynamic for runtime adaptation. Leveraging on a library of deployable microservices, the Execution Platform allows the instantiation of an Interoperability Layer and an Autonomic Layer. The Interoperability Layer is composed by the set of microservices which are responsible for the communication and semantic adaptation to the available IoT Devices and/or External Services, hence performing the Monitoring of the environment and collection of data, and Execution of specific actions to implement the decision taken from the Autonomic Layer. The Autonomic Layer is composed by the set of Smart Behaviours microservices which are responsible for Analysing data and consequently Planning the set of actions for satisfying the requirements of the modelled application logic. The Modelling & Validation Framework offers the ability to use development-time models to supervise running Execution Platform states. This solution enables monitoring of microservices states and starting and stopping them from the Modelling & Validation Framework. BRAIN-IoT Platform is also able to use models@runtime to modify the system's behavior at runtime in response to changes within the system.

3.2 Objectives

To address the challenges outlined in Section 1, BRAIN-IoT platform aims to support IoT professionals to reducing the effort for designing, developing, deploy, configure, optimize, and maintain IoT Systems based on existing and new IoT Services. BRAIN-IoT simplify the process of implementing and deploying code into production, i.e. scaling, capacity planning and maintenance operations may be hidden from the developer or operator. To pursue this ambition, BRAIN-IoT aims to achieve the following four macro-objectives, focusing on two sub-objectives each.

I. *Facilitates the specification and design of complex IoT systems*.

A. Defining a Domain-Specific Modelling Language for IoT and CPS service composition enabling the creation of mashups of existing and new IoT services communicating with different protocols.

B. Specifying the data-flow mappings between the different IoT service interfaces in a seamless way, as well as the behaviour logic implementing the orchestration of such services.

II. *Enabling self-adaptive deployment and management of distributed IoT systems*.

A. Developing a dynamic Cloud/Edge runtime infrastructure which simplifies and automates the process of deploying and managing distributed IoT applications, allowing the automatic installation and replacement of smart behaviours, as well as semantic and syntactic adapters for IoT Devices and Services, reacting to environmental changes and User events.

B. Adopting advanced self-learning features, realized by means of modular Artificial Intelligence algorithms, which can recognize non-critical and critical events (such as data stream anomalies) to trigger the reconfiguration (such as new behaviors deployment, self-healing capabilities, etc.) of the runtime infrastructure.

III. *Enforcing Security and Privacy*.

A. Establishing Authentication, Authorization and Accounting (AAA) in dynamic, distributed IoT scenarios.

B. Providing solutions to embed privacy-awareness and privacy control features in IoT solutions.

IV. *Complex IoT systems validation and safety enforcement*.

A. Developing IoT devices models for evaluating the system in a safe environment before executing it in the real environment.

B. Fast prototyping and supervising of critical running systems, via real-time models, to validate them before instantiating in the real environment.

3.3 Architecture

To achieve its objectives, BRAIN-IoT project develops a Platform which consists of four main technological categories:

Modelling & Validation Framework. Defines a domain-specific Modelling Language describing IoT devices capabilities and system-level behaviours; provides toolset supporting the syntax of the modelling language, allowing model verification and model checking, and automatic code generation to provide rapid model-based development approach.

Execution platform. Provides an autonomic distributed infrastructure for the dynamic deployment and execution of smart behaviours and IoT devices and external services adapters.

Security & Privacy Framework. Provides security and privacy awareness and protection throughout the BRAIN-IoT platform, including end-to-end security service.

Smart Behaviours. Enable autonomic functionalities for the Runtime Infrastructure through the deployment and execution of reusable software AI/ML algorithms.

The four technological categories are made of several software components, or “Products”, which compose the overall BRAIN-IoT Platform, as represented in Figure 2.

In BRAIN-IoT Modelling & Validation Framework a new meta-language, namely the BRAIN-IoT Modelling Language (BRAIN-IoT-ML) is defined. BRAIN-IoT-ML is a UML profile, which extends the generic UML for the IoT domain. BRAIN-IoT-ML implements the concepts in the IoT-A (De Loof, et al., 2013) domain model.

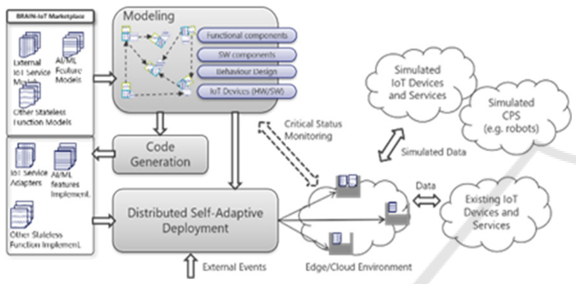
BRAIN-IoT Marketplace
External IoT Service Models
AI/ML Feature Models
Other Stateless Function Models
IoT Service Adapters
AI/ML features Implement.
Other Stateless Function Implement.
Modeling
Functional components
SW components
Behaviour Design
IoT Devices (HW/SW)
Code Generation
Distributed Self-Adaptive Deployment
External Events
Critical Status Monitoring
Simulated IoT Devices and Services
Simulated CPS (e.g. robots)
Simulated Data
Data
Existing IoT Devices and Services
Edge/Cloud Environment

Figure 2: BRAIN-IoT Platform Architecture.

BRAIN-IoT-ML modelling approach is component-based and behaviours are described as state-machines. As a UML profile, BRAIN-IoT-ML integrates well with other UML profiles, e.g., MARTE, SysML, and AI Module profiles to be developed within the project.

IoT-ML aims at proposing a system-level description of functional models. The goal is to add IoT domain concepts in this kind of model, at this level of abstraction. AI concepts shall also complement the IoT models and allow syntactical and semantical compatibility analysis between AI modules.

BRAIN-IoT-ML and AI Module Modelling Language have a dedicated modelling tool to apply Model-Based Engineering. BRAIN-IoT-ML modelling tool uses a Model-Based Engineering (MBE) approach to help component-based modelling, facilitating the linking toward real devices and external services through meta-data representation in WoT TD (W3C, 2020), monitor IoT state-machine based behaviours in a human-friendly graphical manner through the models@runtime approach, and finally quickly prototyping before deployment in the real context to check the goodness of the designed state machine – leveraging on the human friendly graphical monitoring of the state machine itself, by generating the source code for monitoring and controlling the BRAIN-IoT adaptation microservices. The tool also validates syntactical and semantic compatibility between AI modules assembled together.

Since the goal of BRAIN-IoT Platform is not to re-implement yet another IoT middleware but to facilitate the management of complex IoT systems, the BRAIN-IoT adaptation microservices have not been implemented from scratch; counterwise a mature existing IoT platform such as Eclipse sensiNact (Eclipse, 2020) has been integrated to cope with the interoperability issues.

Eclipse sensiNact consists of a software platform enabling the collection, processing and redistribution of any data relevant to improving the quality of life of urban citizens, programming interfaces allowing different modes of access to data (on-demand, periodic, historic, etc.). Eclipse sensiNact IoT devices, legacy systems, mobile applications, open data repositories are the potential exploitable data sources: sensiNact provides connectivity support to those data sources including IoT protocols and platforms such as LoRa, Zigbee, IEEE 802.15.4, Sigfox, enOcean, MQTT, XMPP, NGSI, HTTP, CoAP, etc.

The fully modular BRAIN-IoT Execution Platform is designed not only to be autonomous in its own right, in that it will be entirely self-managing, and self-healing but also it will facilitate the deployment of autonomous microservices, irrespective of the functionality of that microservice. Microservices have the capability of dynamic deployment throughout the framework. In a use case with variable and unpredictable numbers and locations of active end devices, this ability is not available today. Any new edge device on the framework, irrespective of location, will be able to dynamically load microservices as required, given the appropriate security permissions (Nicholson, et al., 2019). Similarly, reconfiguration of the microservices will be possible whenever needed. The current state of the art requires that the overall system or application is stopped, a configuration pushed out to the edge devices and the system is then restarted. The BRAIN-IoT Platform will permit dynamic, “on the fly” reconfiguration and upgrade without necessarily halting the total system. Due to the modularity of the framework and its components, microservices available to the framework can be reused or repurposed for other applications with minimal effort, cutting down both the cost and implementation time of new applications.

Along with the support of Smart Behaviours functionalities, implemented as AI/ML algorithms, the BRAIN-IoT Platform includes the ability for designed applications running on the platform to self-optimise. This is achieved by embedding the ability to monitor the surrounding environment and the operating context via specific adaptation microservices, and then to make analysis and planning with machine learning (ML) to determine the best possible configuration of microservices for achieving the application objectives and requirements. Autonomous dynamic deployment built into the BRAIN-IoT Platform means that microservices can, if deemed appropriate by such an intelligent management agent, be relocated within the available infrastructure. This totally autonomous capability within the BRAIN-IoT Platform represents a significant advance on current state of the art in IoT implementations, paving the way for the Next-Generation IoT paradigm.

4 SERVICE ROBOTICS CASE STUDY

The overall depicted concept draws requirements and challenging use cases from IoT applications in a Service Robotics usage scenarios, which provide the suitable setting to reflect future challenges in terms of dependability, need for smart behaviour, and security, which are expected to become more significant and impacting in the long-term (10+years).

The Service Robotics use-case involves several robotic platforms, like the open-source Robotics Operating System (ROS), which need to collaborate to scan a given warehouse and to assist humans in a logistics domain.

The use-case identified for testing the BRAIN-IoT Platform consists in five robots moving autonomously within the warehouse, with the goal of picking carts from a loading area and moving them to an unloading area. While performing this task, modelled via BRAIN-IoT-ML and deployed as software artefacts onto the different robots, the robots may run into several issues, such as system failures, battery depletion or cyber-attacks.

The role of BRAIN-IoT Platform is to enforce a self-healing behaviour which will guarantee the system to accomplish the full execution of the task. This is achieved implementing a reusable and generic component that intends to abstract the implementation of anomaly detection of a specific problem to generalize it to problems of the same type.

Aims to identify the anomalies over several data sources, that can be due to a change in the behaviour of the sensor, a failure or a miscalibration. It is necessary to specify the main data sources where the detection of anomalies shall be done and a set of co-variables, if any. The Communication between adaptation microservices – gathering data from data sources – and such Smart Behaviour microservice is implemented through the BRAIN-IoT Event Bus. The Brain IoT Event Bus is a lightweight, asynchronous eventing system designed to allow communication between decoupled resources and software components. It provides type safe access to data, and basic schema transformation. This allows components to interact even if their data schema is not known until compile time.

The event bus is also responsible for interacting with various security components to validate the integrity and origin of data as it flows through the system. This includes the use of the authorisation engine to limit the sending and delivery of event data.

An analysis of the input data is carried out. It attempts to identify the main characteristics of each of the data sources and the correlation between the different sources. According to the analysis, the variables to be modelled are selected and a set of complementary methods is executed, which by different techniques, extract different types of anomalies. This anomaly detection is translated in an event which triggers the instantiation of the same logic to a new robotic platform which was not part of the original set of robots and can substitute the robots which presents the detected anomaly. The Execution Platform provides automated distributed management of microservices placement, based upon the concept of "unhandled events" and the BRAIN-IoT artefact repository. Microservices and behaviours are dynamically "resolved" by a Behaviour Management Service, implemented within the Execution Platform, on the target node to validate that their dependencies can be satisfied, and that the node is capable of running them. Such Behaviour Management Service takes the role of an OSGi Management Agent (a role identified, but not specified, by the OSGi Alliance) and is responsible for the deployment and management of OSGi bundles and configurations in the runtime. As such it makes use of related OSGi specifications such as the Resolver Service and the OSGi Repository Service. The concepts used by the Behaviour Management Service are relatively common in OSGi, although the use of the resolver service to determine deployments at runtime is used less frequently. The primary innovation is the use of live event data from the event bus to dynamically trigger the installation of microservices and behaviours.

5 CONCLUSIONS

This paper has presented the BRAIN-IoT platform, explaining how this solution paves the way for the development and demonstration of novel IoT concepts and solutions, supporting the Next Generation Internet of Things vision. In the next years, such vision, will require the introduction of next generation IoT architectures able to autonomously react to system issues and adopt self-healing and self-protection techniques to guarantee the reliability and service continuity of complex distributed IoT systems.

The BRAIN-IoT project provides a platform that aims to be considered as a reference solution for the development of applications involving autonomic computing and distributed IoT systems. The solution will provide both a Modelling and Validation Framework that allow designing applications using a modern model-based design approach and an Execution Platform, which implements a MAPE loop. The main features provided by this platform in terms of dynamicity and self-* capabilities are possible thanks to the use of the OSGi-based technologies and more specifically the Requirements and Capabilities specification, which allows to handle the autonomous dynamicity of the platform as an automatic dependencies matchmaking. Leveraging such technology, the BRAIN-IoT delivers a software eco-system capable of deploying/assembling, orchestrating and managing sophisticated IoT applications. Conceptually, like, but significantly more flexible than, compute lambdas; BRAIN-IoT demonstrates how sophisticated distributed behavioural pipelines of software components can be dynamically assembled, in response to environmental triggers. Meanwhile, by pursuing a modular structural hierarchy (i.e., an holonic approach), BRAIN-IoT enables the natural creation of federated distributed environments ideally suited to the demands of several domains of the modern society.

6 FUTURE WORKS

The BRAIN-IoT project use cases demonstrate the flexibility of the proposed approach within a diverse set of operational environments. Specifically, this paper has focused on the Service Robotics use case, which demonstrates the use of BRAIN-IoT platform in a highly automated, dynamic and adaptive manufacturing environment. Furthermore, the same approach will be applied also in the second use case concerning the monitoring and control of public water management infrastructure for the city of A Coruna, which will demonstrate the use of BRAIN-IoT in critical infrastructures and mission-critical environments.

Finally, during the final phase of the BRAIN-IoT project, key BRAIN-IoT concepts will be finalized with regards with their OSGi Alliance specification process (OSGi Github, 2019).

ACKNOWLEDGEMENTS

This work was supported by the project “BRAIN-IoT” (H2020-IOT-03-2017, grant agreement no. 780089).

REFERENCES

Abeywickrama, D. B., & Zambonelli, F. (2012, April). Model checking goal-oriented requirements for self-adaptive systems. *IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems*, pp. (pp. 33-42).

Azimi, I., Anzanpour, A., Rahmani, A. M., Pahikkala, T., Levorato, M., Liljeberg, P., & Dutt, N. (2017). HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT. *ACM Transactions on Embedded Computing Systems (TECS)*, pp. 16(5s), 1-20.

Beal, J., Pianini, D., & Viroli, M. (2015). Aggregate programming for the internet of things. *Computer*, pp. 48(9), 22-30.

Bencomo, N., Götz, S., & Song, H. (2019). Models@ run. time: a guided tour of the state of the art and research challenges. *Software & Systems Modeling*, pp. 18(5), 3049-3082.

Blair, G., Bencomo, N., & France, R. B. (2009). Models@ run. time. *Computer*, pp. 42(10), 22-27.

Brain-IoT. (2018). [Online]: http://www.brain-iot.eu/

Bucchiarone, A., Marconi, A., Pistore, M., & Raik, H. (2017). A context-aware framework for dynamic composition of process fragments in the internet of services. *ournal of Internet Services and Applications*, pp. 8(1), 6.

Bueno, L. C. (2012). *A Reference Architecture for Component-Based Self-Adaptive Software Systems.* Department of Information and Communication Technologies Faculty of Engineering, ICESI University, Cali, Columbia.

De Loof, J., SAP, C. M., Meissner, S., Nettsträter, A., CEA, A. O., SAP, M. T., & Walewski, J. W. (2013). *Internet* *of Things–Architecture IoT-A Deliverable D1. 5–Final architectural reference model for the IoT v3. 0.*

Eclipse. (2020). [Online]: https://projects.eclipse.org/proposals/eclipse-sensinact

Ferrera, E., Pastrone, C., Brun, P. E., De Besombes, R., Loupos, K., K. G., & Mygiakis, A. (2018). IoT European Security and Privacy Projects: Integration, Architectures and Interoperability. Next Generation Internet of Things. Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation.

Fettweis, G. P. (2014). The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1), 64-70.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1), 41-50.

Lee, E., Seo, Y. D., & Kim, Y. G. (2019). Self-Adaptive Framework Based on MAPE Loop for Internet of Things. *Sensors*, 19(13), 2996.

Muccini, H., Spalazzese, R., Moghaddam, M. T., & Sharaf, M. (2018, September). Self-adaptive IoT architectures: An emergency handling case study. *In Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pp. (pp. 1-6).

NGIoT, N. G. (2019). *Building a Roadmap for the Next Generation Internet of Things – Research, Innovation and Implementation 2021-2027.* [Online]: https://ngiot.eu/wp-content/uploads/sites/26/2019/09/NGIoT_scoping-paper.pdf

Nicholson, R., Ward, T., Baum, D., Tao, X., Conzon, D., & Ferrera, E. (2019, July). Dynamic fog computing platform for event-driven deployment and orchestration of distributed Internet of Things applications. *Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*, pp. (pp. 239-246). IEEE.

OSGi Github. (2019). [Online]: https://github.com/osgi/design/blob/master/rfcs/rfc0244/rfc-0244-Type-Safe-Events.pdf

Ouechtati, H., Azzouna, N. B., & Said, L. B. (2018, January). Towards a self-adaptive access control middleware for the Internet of Things. *In 2018 International Conference on Information Networking (ICOIN)*, pp. (pp. 545-550). IEEE.

Pianini, D., Montagna, S., & Viroli, M. (2013). Chemical-oriented simulation of computational systems with Alchemist. *Journal of Simulation*, pp. 7(3), 202-215.

Renart, E. G., Diaz-Montes, J., & Parashar, M. (2017, May). Data-driven stream processing at the edge. *In 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp. (pp. 31-40). IEEE.

Renart, E., Balouek-Thomert, D., & Parashar, M. (2017, September). Pulsar: Enabling dynamic data-driven IoT applications. *IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W)* , pp. (pp. 357-359).

Ribeiro, A. D., de Almeida, F. M., Moreno, E. D., & Montesco, C. A. (2016, September). A management architectural pattern for adaptation system in Internet of Things. *In 2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. (pp. 576-581). IEEE.

Sylla, A. N., Louvel, M., Rutten, E., & Delaval, G. (2017, September). Design framework for reliable multiple autonomic loops in smart environments. *In 2017 International Conference on Cloud and Autonomic Computing (ICCAC)*, pp. (pp. 131-142). IEEE.

Vermesan, O., & Bacquet, J. (. (2019). *Next generation Internet of Things: Distributed intelligence at the edge and human machine-to-machine cooperation.* River Publishers.

Villegas, N. M., & Müller, H. A. (2011). *Context-driven adaptive monitoring for supporting SOA governance. CMU/SEI-2011-SR-008.* Pittsburgh: Carnegie Mellon University.

W3C. (2020). [Online]: https://www.w3.org/TR/wot-thing-description/

Welsh, K., Bencomo, N., Sawyer, P., & Whittle, J. (2014). Self-explanation in adaptive systems based on runtime goal-based models. *In Transactions on Computational Collective Intelligence XVI*, pp. (pp. 122-145). Springer, Berlin, Heidelberg.

Zhang, L., Alharbe, N., & Atkins, A. S. (2016, December). An IoT application for inventory management with a self-adaptive decision model. *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. (pp. 317-322). IEEE.