

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:

2021-09-02

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Rahmouni, A., Silva, C. F. & Ghodous, P. (2020). Methodological approach for creating an IoT manufacturing application. In Boubchir, L., Benkhelifa, E., Jararweh, Y. and Saleh, I. (Ed.), 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). Paris, France: IEEE.

Further information on publisher's website:

10.1109/IOTSMS52051.2020.9340188

Publisher's copyright statement:

This is the peer reviewed version of the following article: Rahmouni, A., Silva, C. F. & Ghodous, P. (2020). Methodological approach for creating an IoT manufacturing application. In Boubchir, L., Benkhelifa, E., Jararweh, Y. and Saleh, I. (Ed.), 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). Paris, France: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/IOTSMS52051.2020.9340188>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Methodological approach for creating an IoT manufacturing application

Achraf Rahmouni
LIRIS UMR 5205
Université Claude Bernard
Lyon, France
achraf.rahmouni@liris.cnrs.fr

Catarina Ferreira da Silva
Instituto Universitário de Lisboa
(ISCTE-IUL), Centre for Informatics
and Systems of the University of
Coimbra
Coimbra, Portugal
catarina.ferreira.silva@iscte-iul.pt

Parisa Ghodous
LIRIS UMR 5205
Université Claude Bernard
Lyon, France
parisa.ghodous@liris.cnrs.fr

Abstract— With the heterogeneity and diversity of electronic systems and their memory capacity, the Internet of Things technologies face several challenges, such as the difficulty in choosing in a growing catalog of IoT technologies and the lack of interoperability of these technologies. This leads to the creation of particular IoT architectures and to the selection of IoT technologies by trying and error approaches. We propose a methodological approach for creating an IoT application guided by the users' needs and by the contextual environment. It enables the user to select the right hardware guided by the market supply and to check the performances of the IoT solutions. We implement and test our proposition in a picking system of a manufacturing factory. This practical implementation provides feedback regarding our methodological approach for creating an IoT application, which in turn enables us to improve and remodel the approach.

Keywords—internet of things, manufacturing industry, methodological approach, sensors, actuators, system on chip

I. INTRODUCTION

IoT allows the creation of multiple services that can, for example, monitor the environment and the weather, detect events, trigger actions, support decision-making and improve the quality of life by automating everyday tasks [1]. Many IoT applications have already been implemented for industrial automation, collision avoidance system in cars, energy efficiency, smart buildings/houses etc. [2]. Each application domain is associated with their own suitable IoT technologies. These various IoT technologies in today's market generate challenges related with the diversity, complexity, heterogeneity and lack of interoperability of the IoT technologies. A user has to face these challenges when building and implementing an IoT application. This prompts us to seek solutions to adapt IoT applications to those different challenges. In the literature, there is not yet a single architecture that makes it possible to adapt IoT technologies to all fields of application. Each user tends to develop and set up their own architecture based on their own selection of IoT technology. The users implement specific architectures linked to their own needs. This generates a diversity of IoT technologies from which to choose. That's why there is a need for a semi-automated approach that can help those users. Therefore, we propose a methodological approach to guide the users in the design and implementation of their IoT application's architecture. It intends to help the user to choose the relevant IoT technologies and design a suitable IoT application architecture. Our approach is based on a set of guiding steps to follow.

In the remainder of the paper we present some existing IoT methodology, followed by an analysis of IoT architectures'

design, we then propose our methodological approach to create an IoT application, followed by an empirical implementation and validation in a manufacturing industry domain. We also present an evaluation and discussion. Conclusion and perspectives close the paper.

II. EXISTING METHODOLOGIES

In this section we present some methodologies in industry 4.0 to design an IoT application scenario. Our research is on this domain since our methodology is tested in a factory that wishes to integrate the IoT in their installations (section V).

A. Design principles for Industry 4.0 scenarios

In industrial field, a study was carried out by Hermann and colleagues [3] to design scenarios within factories and propose a design principle to provide practical guidance for the design of IoT applications. The design of the industrial scenarios follows five main steps: (1) Create a common understanding of industry 4.0 design principles, based on discussion by the participants, (2) Identify and specify industry 4.0 scenarios: the project's participants put together a list of technologies made from an analysis of industry 4.0 applications installed in others companies, (3) Evaluate the identified scenarios using the analytical hierarchy process (AHP) approach to evaluates the return on investment. This process includes three steps: identification and selection of criteria, quantitative determination of criteria and evaluation of potential scenarios using weighted criteria, (4) Specify and re-evaluate the prioritized scenarios, and (5) Prepare the selected scenarios for implementation. These steps are analyzed in a research project that was carried out in a factory by TU Dortmund University.

B. Methodology for monitoring manufacturing environment

This methodology is adapted in the context of a manufacturing industry that allows the control and monitoring of real-time temperature in an office [4]. The proposed architecture is based on the following four layers. Layer 1: data acquisition is done using wireless sensors; Layer 2: the microcontroller collects data, process and manages the system interface for reading and displaying the data processed; Layer 3: the use of a solution in Cloud Computing is recommended. This provides services for data storage and the possibility of hosting servers. With Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS), the server can manage notifications, analyze data while streamlining the system architecture; Layer 4: the development of a software solution is needed to display and monitor data. A mobile application is recommended for more flexibility and accessibility.

The service-oriented methodology presented in our article aims to give the users in the set-up of IoT applications based on the knowledge of one or more people, unlike the step 2 of the methodology developed by Hermann and colleagues [3]. As far as cloud computing is concerned, and since the manufacturing industry has not opted for this technology, the data is locally processed. In addition, the integration of layer 3 of the methodology of Li and Kara [4] remains to be integrated in our future work.

III. IOT ARCHITECTURES

The integration of IoT in industry and building as well as in many other domains of everyday life, is exposed to many problems currently known, such the research of a reference IoT architecture [5, 6, 7]. Among other, the diversity of IoT technologies and constraints represents a major challenge when deciding for a specific architecture to support the implementation of an IoT application. Whitmore and colleagues [8] raise the interest to design an IoT architecture and present some of the issues that need attention, such as: “*design of distributed open architecture with end-to-end characteristics, interoperability of heterogeneous systems, neutral access, clear layering and resilience to physical network disruption, decentralized autonomic architectures based on peering of nodes*”. This section presents existing IoT architectures and summarizes the details of three layers, which contextualize our work.

Several works have been proposed in literature regarding IoT architectures to fit their own needs [9, 10, 11, 12]. The IoT technologies are characterized by increasing heterogeneity and diversity. Moreover, to design and to deploy an IoT application, organizing and structuring step are needed so that the IoT application answers the clients’ requirements and works with acceptable efficiency. Most of the proposed architectures cited in literature in [9, 10] are subdivided into several layers. In the industry we can find the RAMI 4.0 [11] and IIRA [12] which intend to support and manage the IoT within the industry. The analysis of IoT architectures in the field of manufacturing industry brings us to consider the three main layers on which an IoT application depends (Table I). Therefore, our methodology depends on this structure. A comparative table is set up to show the requirements of our approach (Table I). The requirements *Identification of the area of application* and *Evaluation of the specificities and the requests of the client* cross-cut all layers. The physical layer contains all electronic systems namely sensors, actuators and System on Chip (SoC).

TABLE I. MAIN REQUIREMENTS REGARDING THE DIFFERENT LAYERS OF THE ARCHITECTURE

The physical layer	The network layer	The application layer
Identification of the area of application Evaluation of the specificities and the requests of the client		
Checking capacity and multitask characteristics of the processor Checking the hardware overcrowding	Selection of the communication network	Testing the selected technologies Setting up the IoT application
Choosing SoCs Choosing actuators Choosing sensors		

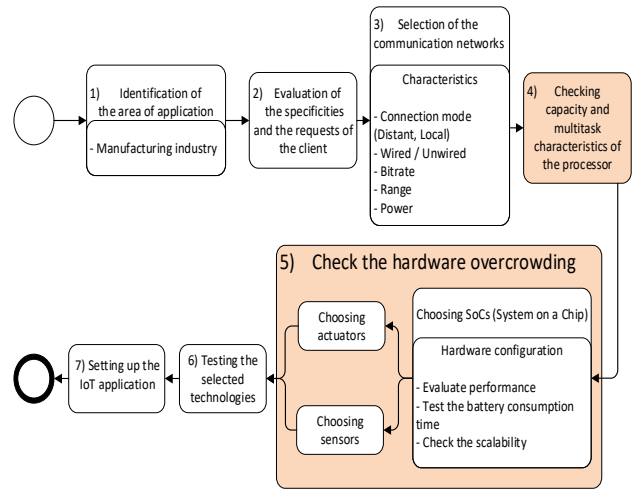
The network layer groups together all the communication networks used to transfer data between the physical layer, the gateway and the workstations. The application layer allows the

user or other applications to interact with the IoT devices’ data and can analyze the parameters provided by the physical layer. According to this analysis, the user or the other applications can control those devices. The steps “*checking capacity and multitask characteristics of the processor*” and “*checking the hardware overcrowding*” are explained in the subsections V.A and V.D.

IV. METHODOLOGICAL APPROACH TO CREATE AN IOT APPLICATION

The methodological approach we propose to create an IoT application is composed of the following steps (Fig. 1): 1) Identification of the area of application; 2) Evaluation of the specificities and the requests of the client; 3) Selection of the communication networks; 4) Checking the capacity and the multitask characteristics of the processor; 5) Selection of actuators, sensors and SoC (system on chip), taking into account the hardware over-crowding and configuration; 6) Testing the selected technologies, and finally, 7) Setting up the IoT application.

Fig. 1. Our Methodological Approach to Create an IoT Application.



Our methodological approach was devised and refined in several iterations, driven by an implementation on a manufacturing picking system. This empirical implementation and validation are detailed in section V. The IoT application provides a service enabling to improve the picking system of manufacturing pieces. After two iterations of creation, implementation and testing of the connected objects, we needed to improve and remodel our initial proposition of our methodological approach. As so, we deemed important to add the tasks *checking capacity and multitask characteristics of the processor* and *checking the hardware overcrowding* to our methodological approach. These new tasks are colored in orange in Fig. 1 (steps 4 and 5). In the subsequent subsections we explain all steps of our methodological approach.

A. Identification of the area of application

The identification of the area of application is the first step of our approach, because depending on the application area, the selection of the IoT technologies and devices may vary.

TABLE II. TWO DIFFERENT IOT APPLICATION DOMAINS AND CORRESPONDING USE CASE EXAMPLES

Application domain	Use cases	References
Industry 4.0	<ul style="list-style-type: none"> Monitor leaks of polluting gases Locate employees Improve the manufacturing process 	[13, 14, 15]

Application domain	Use cases	References
Smart home/ Smart building	<ul style="list-style-type: none"> Control electronic devices Improve security Manage energy consumption and the lighting Adapt connected objects to the user's habits 	[16, 17]

This step will affect the selection of technologies as well as for all the steps of the methodology and especially the evaluation of the specificities (step 2) which will consider the customer requirements when considering adding an IoT application in an old system already in place. Table II summarizes a set of use case examples. It intends to contextualize our approach, as different application domains frequently require different IoT technologies. Generally, IoT applications are based on four main aspects: monitoring (e.g., device status, environmental status, notifications, alert), control (device functions), optimization (e.g., device performance, diagnostics, repair) and autonomy (autonomous operations) [2].

B. Evaluate the specificities and the requests of client

After selecting the application domain, the evaluation of the specificities and of the requests of the client is guided by the answer to the two following questions:

1) *What is the deployment environment of our application?* The study of the environment makes it possible to better choose the IoT technology. In an industrial environment, each workspace can combine a maximum of production means. In some factories, this space is generally divided into work zones. Each work zone plays a sequential production role that consists of doing a particular task following a well-defined order to have a final product that match with the client's requests.

By doing this study we can make a diagnosis of the place where IoT will be installed and thus take measures to choose the right object according to the size of the equipment that must fit with the installations already set up by the client.

2) *Can the IoT application adapt to the system already in place?* Generally, factories already have a system of management and control of different devices. Which lead us to study and see more closely the technologies used, regarding the hardware side (sensors, actuators), and the software side (communication networks, communication protocols, platforms etc.). Before developing an IoT application, we should check the programming language used in order to make the application interoperable and follow the standards imposed by the manufacturers (regarding security, cost, ...), the customer requests (namely quality, efficiency ...) and other possible constraints.

C. Selection of communications networks

Network criteria determines the choice of IoT applications. The characteristics of the network have an impact on the subsequent selection of IoT components. The main network criteria to consider are network range, data bit rate, battery life, and network cost. They are divided according to their categories, wireline, proximity, WPAN etc. and classified in order of range: from wired networks to GSM networks (> 350m).

Delaney and colleagues [19] present a framework which can give an automatic choice of a communication network that adapts to the user's IoT application. The user presents the requirements of their IoT application, and the software

chooses an appropriate network solution. This framework is based on predefined models to establish the expected performance of a solution in a given environment. These models are based on data acquired from the tested solutions. The design of this framework is based on four principles: (1) Maintaining a repository of solutions. (2) Solution modeling. (3) Deployment and analysis. (4) Selection and deployment of the solution. The purpose of these tests is to present comparisons between communication networks by following evaluation measures. These measures depend on four criteria: reliability, latency, efficiency and stability. Based on those measurements, tests were carried out on the TinyOS SIMulator (TOSSIM) made by Levis and colleagues [20, 21]. The proposed software provides tests and selection of the right network.

Battery life is a challenge for mobile IoT application that remains to be solved. The energy consumption is often linked to the number of devices, the communication network protocol and the data rate. Morin and colleagues [22] give an analysis to calculate the energy consumption and lifetime of wireless communication network technologies. This calculation follows the work of Vilajosana [23] to express the energy consumption in a time interval t . Following this, an algorithm was made to calculate the energy consumption and therefore the lifespan of the device. Tests were carried out with SoCs equipped with a microcontroller and a communication network. These devices are powered by two AAA batteries (1250mAh at 1: 5V each) which store an energy capacity of the order of $E_0 = 13 \text{ kJ}$ initially full.

Network cost varies according to the data size and the bitrate required by the IoT application. The tariffs for sending a few messages a day via the GSM network and the low frequency technologies are expensive. The cost of a 3G or 4G cellular modem is higher than a modem LPWAN. To use conventional cellular networks to transmit a few kilobytes per object per day can result in disproportionate expenditure when scaling up massive deployment. Mekki and colleagues [18] show different costs of LPWAN technologies by comparing Sigfox, LoRa and NB-IoT networks.

In some cases, the selection of communication networks is made in advance. It depends on the technologies used in the system already in place. In most cases the customer imposes his own communication network which can cancel this step of the methodology and move on to the next step.

D. Checking capacity and multitask characteristics of the processor

After performing a first iteration of our methodological approach (Fig. 1 without the orange colored boxes) to create an IoT application and then testing it (details are presented in section 5), we verified the second connected object and found that he was a single task processor. This was insufficient, knowing that the client manufacturing work zones are equipped with at least 4 trolleys and need to work with multitask characteristics. This led us to improve our design approach. We need to consider the verification of the characteristics of the client manufacturing work zones, in particular if they imply the selection of multitask processors. As so, we added the task box *check the capacity and multitask characteristics of the processor* (improvement of step 4) to our approach.

E. Hardware selection : SoC, sensors and actuators

1) *System on chip (SoC)*: To operate sensors and actuators, an electronic card is required in which data processing will take place. The electronic card or SoC

operating system allows to easily develop and integrate programs able to control the sensors and actuators answering the client's needs. A lot of SoC are available in the current market. The SoC performance differs according to the RAM memory, CPU frequency, GPU memory, supported programming languages, and network communication protocols.

2) *Sensors*: A sensor is a technology that relies on an electronic system that can capture a physical quantity from the real world and turn it into an electrical signal. After processing this signal, it can be transmitted to a computerized system to read this data. The sensors have the advantage to collect data continuously. They are designed to accomplish tasks such as monitoring temperature, humidity, movement and mass. Sensor devices can work cooperatively and form a network of sensors.

3) *Actuators*: An actuator has the capability to convert a command from a computer system into a mechanical action that will affect its environment installation in order to make a change in the real world. The types of actuators differ according to their power and speed. We can find three types of actuators: (1) Hydraulic actuators rely on a fluid as a main source of energy such as water, oil etc. (2) Pneumatic actuators rely on air compression. (3) Electric actuators rely on AC and DC motors or stepper motors. Actuators can create linear motion, rotary motion, or oscillating motion.

When implementing our approach in the manufacturing empirical implementation (section V) we found the need to consider the overcrowding implied by the hardware (SoC, sensors or actuators) and the physical material supporting this hardware. Steps 6 and 7 of our methodological approach are explained in section V.

F. The servitization process

Our methodological approach provides a guidance service to the user through steps from identifying the application area to setting up an appropriate IoT software application. It also helps to select the suitable IoT technologies and to design a high-performance architecture meeting the user's requirements for the IoT application to be set up by using the appropriate devices and materials, and by verifying performance. As a result, our approach provides a servitization process [24, 25], which explains the transition from manufacturing industry offering complex engineered products to services in order to provide solutions that support or complement their products. This transition is based on the following five criteria [24]: (1) the shift from manufacturing raw products to products comprising solutions, (2) from outputs to results, (3) from transactions to relationships, (4) from suppliers to network partners, and (5) from elements to concentrates. As a result, the design of the products assembled at the end of the deployment of our approach, e.g. with regard to SoCs, sensors and actuators generate services to facilitate the operators' work in factories for a better productivity with less routine work. The validation of this approach is explained hereafter.

V. EMPIRICAL IMPLEMENTATION AND VALIDATION

In this section we validate our approach, by showing our intervention in a manufacturing factory with the goal of improving the manufacturing factory picking system with an IoT application. The manufacturing picking system aims at assisting the operator in the order picking operation. Order picking usually consists of grouping together several products of different types to prepare them before assemblage. This

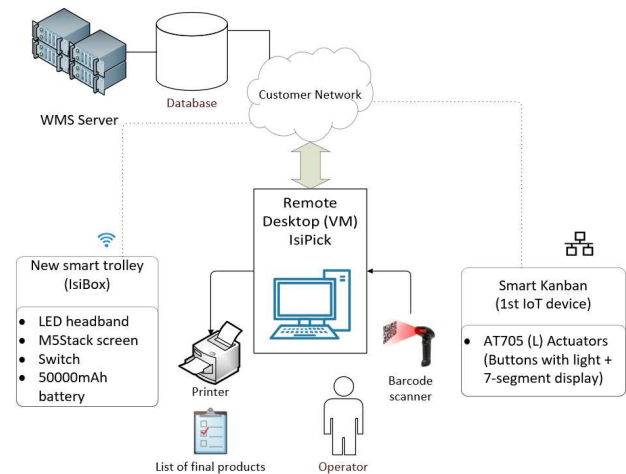
process relies on various sensors, actuators and electronic cards that have to be selected and combined in order to have a complete service capable of providing the necessary features to satisfy the needs of the client (i.e., the manufacturing factory) and specially to relieve the work of the operators.

Fig. 2. TCP/IP Controller, Router, Battery, an AT705(L) and 4 AT70N; Trolley.



Following our approach, a first IoT was installed and it contained a module available for sale by ATOP Technologies. This module consists of a large light button, a 7-segment display and two small buttons on the left of the screen (Fig. 2). This module is connected to a TCP/IP controller (black box on top of Fig. 2). The controller is connected to a router (white box) and enables Wifi-RS232 or Ethernet-RS232 network. This IoT module is installed on the Kanban furniture (were the factory products are placed) and on the trolley (Fig. 2 right side) which is powered by a battery of the order of 50000 mAh.

Fig. 3. Architecture of the IoT application installed in the manufacturing factory.



Before the installation of our IoT application, the order picking was done manually. The operator had to print the orders, manually search the products by reading their labels in the warehouse and then deposit them in the required place. After the installation of our IoT application, for a given command, the operator scans a barcode associated with this command using a barcode reader (Fig. 3). This barcode reader is connected to the workstation and sends the reading information to the IsiPick software to prepare the desired picking. The IsiPick software controls the modules already installed on the Kanban furniture and on the trolleys. During a sampling operation, the light buttons installed on the Kanban and the 7-segment display will light together in a specific order. These light buttons have the role of showing the location of the products. As for the 7-segment displays will show the number of products to be sampled. This order is calculated by the IsiPick software. IsiPick will calculate the picking orders's products, to build the product customer's request. This picking order allows the operator to carry out this operation without worrying about the problems of movement

in the warehouse. IsiPick will also lighten the work for the operator by making a minimum of trips in the warehouse. On the other hand, IsiPick allows to communicate the required information to light the buttons and the 7-segment display either on the Kanban furniture and on the trolley.

In the following subsection, we present the problems we found in this first version of the IoT application.

A. The problems found with the first IoT application and design of a second one

The first version of the IoT application in the client manufacturing factory reveals the following facts: (1) The IoT solution depends on several components and it is necessary to provide a special box to group and form a compact module that protects them in case of shocks and allow their installation on the trolley and the Kanban. Also, it must be both lightweight for the operator and space saving, knowing that it is the operator who pushes the trolley, by her own means, when the automatic guided vehicle (AGV) is in use in another section or in maintenance; (2) The size of the box must be adjusted before its installation on the Kanban and the trolley: its position must not congest the operator in their work; (3) The lack of interoperability: ATOP's products can only be operated from their manufacturer's own program that relies on the client's DLL libraries which limits the selection of IoT technology. These problems led us to design a second connected object, more compact and with the intention to replace the ATOP IoT components that lack interoperability and are overcrowding.

The second connected object is formed of a Wemos SoC. This SoC has a Wi-Fi network and an ESP8266 microcontroller. The ESP8266's role is to read or send the frames. The Wemos card gives us the possibility to replace the controller and the router, which reduces the number of components. In order to replace the light buttons, we introduce in a banner of four LEDs. Each LED is associated with a trolley position. Finally, we replaced the 7-segment display with 5 digits of the AT705 (L) module by a 7-segment display with 8 digits. These electronic components have been grouped in one box to save space. This box contains the Wemos electronic board, a button, a 7-segment display with 8 digits, a 4 LED strip and a battery. However, after several tests, we noticed that the second connected object only worked for one user. The cause of this problem is related with the ESP8266 microcontroller since the IsiPick software, with which it should operate and that sends system errors. The Wemos microcontroller is a single task one and this explains the overflow errors, when it receives more than one command to process. Single task microcontroller is insufficient for our client work zones knowing that each one is equipped with at least 4 trolleys. Also, the four selected LEDs are not very visible while they are on their maximum intensity. This may disrupt the operators. As a result, the design of a new connected object with a microcontroller capable of supporting multiple tasking and LEDs improved visibility become a necessity.

In the next subsection we show how we apply the final version of our methodological approach to create a third version of the connected object for the same client.

B. Design and set up of a third IoT connected object

In order to help the operator with the picking system in the storage warehouse we need different electronic components. These electronic components must inform the user on the position of the desired products, the quantity to take and the position of deposit of these products on the trolley. The IoT

components can be made of a button, a display, LEDs, a battery and the IsiPick software.

1) *Evaluate the specificities and the requests of the customer:*

a) The installation environment of the IoT application is in a car door production factory. This poses no constraints on the choice of the IoT technologies.

b) The IoT application should be deployed on trolleys of the company. It is an isolated system that must be provided with a battery to ensure the operation of our IoT application.

c) The client imposes the Wi-Fi communication network. As a result, the choice of devices must be compatible with this network.

We need to consider the trolley should be able to be pushed by the operators during their work. This drives us to choose components that are not bulky and not heavy.

2) *Choosing the SoC:* The choice of the SoCs are based on the following criteria

a) *The communication network:* In order to ensure the Wi-Fi communication of the trolley in the warehouse of the client's company, an attempt was made to find an adequate microcontroller with a suitable Wi-Fi range. Also, to prevent the problem of insufficient multitasking, our choice fell on microcontrollers with two cores. We selected a dual core microcontroller with an ESP32, which is on the M5Stack SoC. The ESP32 is a standalone Wi-Fi network solution and supports various recent 802.11b/g/n network protocols. In addition, it has a correct Wi-Fi range in the order of 25db and can cover 100 meters. By doing tests in the warehouse, we found that the Wi-Fi connection is continuous.

b) *Performance evaluation:* The market provides SoCs that differ in terms of performance regarding RAM memory, CPU frequency, GPU memory and different types of connectivity network. The M5Stack card has 512kb SRAM and a 240Mhz dual core CPU characterized by its multitasking features which can send and receive tasks of commands in real time. The M5Stack card does not have a graphic card. The screen depends on the processing of the ESP32 microcontroller to display the messages sent by the IsiPick software. Regarding the programming language, the M5Stack card depends on the C++ Arduino library. This library provides the advantage of implementing programs already tested and verified beforehand.

3) *Choose the battery:* It must ensure a continuous power supply for the operation of our connected trolley supported by the M5Stack card and the electronic components to cover at least one working day. One of the strongest points that pushed us to choose this card is its low battery consumption. With a battery of 50000mAh it can hold up to 6 days of battery life.

4) *Choose the actuators:* The M5Stack has already electronic components previously installed. It is equipped with 3 buttons, a screen and a 150 mAh battery. However, we added a switch in order to replace one of the 3 buttons associated with the M5Stack card to facilitates the picking system operation.

We decided to choose LED headbands in order to have more visibility and thus the operator will save more time, as she can easily locate positions on the trolley. In order to protect this LED headbands, a sheet metal has been designed in which the tape will be glued. This metal sheet is equipped with a piece of white plastic that allows the protection and visibility of the LEDs once lit.

In addition, parts were manufactured, using a 3D printer, in order to protect the edges of the metal sheet and to facilitate their set up on the trolley by means of clamps. This new connected object, called IsiBox, is installed on 24 trolleys (Fig. 4).

Fig. 4. The final installed IoT device (IsiBox) on a trolley.



VI. CONCLUSION AND PERSPECTIVES

We propose a methodological approach to create an IoT application, which we validate in the context of a real manufacturing factory. The purpose of this IoT application is to automate the order picking system of the manufacturing factory by installing connected furniture and trolleys. In this practical case, we needed three iterations of our methodological approach, to provide a final IoT application, which satisfies the client's needs and also enabled us to improve our methodological approach.

Our approach can also serve as guidelines for future creation of IoT applications. It guides the user in the process of making the most suitable choices among the different IoT technologies so she can design an IoT application by using the right hardware, checking performances and comply with the client's requirements.

As for perspective we intend to design and implement a recommendation system based on our methodological approach, which would facilitate even more the process of selecting the suitable technologies to create an IoT application.

REFERENCES

- [1] Weyrich, M., & Ebert, C. (2016). Reference Architectures for the Internet of Things. *IEEE Software*, 33(1), 112–116. <https://doi.org/10.1109/MS.2016.20>
- [2] Rozsa, V., Deniszczewicz, M., Dutra, M., Ghodous, P., Ferreira da Silva, C., Moayeri, Figay, N. (2016). An Application Domain-Based Taxonomy for IoT Sensors. <https://doi.org/10.3233/978-1-61499-703-0-249>
- [3] Hermann, M., Pentek, T., & Otto, B. (2016, January). Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii international conference on system sciences (HICSS)* (pp. 3928-3937). IEEE.
- [4] Li, W., & Kara, S. (2017). Methodology for monitoring manufacturing environment by using wireless sensor networks (WSN) and the internet of things (IoT). *Procedia CIRP*, 61(Supplement C), 323-328
- [5] Z., D. R. G., Nait-Sidi-Moh, A., Durand, D., & Fortin, J. (2014). Internet des objets et interopérabilité des flux logistiques : état de l'art et perspectives, 8. Retrieved from <https://hal.archives-ouvertes.fr/hal-01017395>
- [6] Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Personal*

- Communications, 58(1), 49–69. <https://doi.org/10.1007/s11277-011-0288-5>
- [7] Leading the IoT. (2017). Retrieved from https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf
- [8] Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2), 261–274. <https://doi.org/10.1007/s10796-014-9489-2>
- [9] Madakam, S., Ramaswamy, R., Tripathi, S., Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 03(05), 164–173. <https://doi.org/10.4236/jcc.2015.35021>
- [10] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (n.d.). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Retrieved from www.buyya.com.
- [11] Anderl, R., & Anderl, I. R. (2016). Industrie 4.0-Digital Transformation in Product Engineering and Production Anwendungsszenarien für Industrie 4.0 View project autoADD-Automatisierung der Prozesskette zur kundenindividuellen Additiven Fertigung (www.autoadd.de) View project Industrie 4.0-Digital Transformation in Product Engineering and Production. Retrieved from www.autoadd.de
- [12] Industrial Internet Consortium. (2015). Industrial internet reference architecture. Technical Article. Available online: <http://www.iiconsortium.org/IIRA.htm> (accessed on 29 August 2018).
- [13] Takeshi, Kazuhiro Chisaki, J., & Kawakami, Y. (2016). Optimization of Factory Production Activities by Utilizing IoT. *FUJITSU Sci. Tech. J.* (Vol. 52).
- [14] Jing, H., & Junye, Z. (2014). Study on monitor system of pollution discharge in chemical enterprise based on internet of things. *Journal of Chemical and Pharmaceutical Research*, 6(7), 1796–1801. Retrieved from www.jocpr.com
- [15] Lin, K., Wang, W., Bi, Y., Qiu, M., & Hassan, M. M. (2016). Human localization based on inertial sensors and fingerprints in the Industrial Internet of Things. *Computer Networks*, 101, 113–126. <https://doi.org/10.1016/J.COMNET.2015.11.012>
- [16] Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., & Lung, C.-H. (2013). Smart Home: Integrating Internet of Things with Web Services and Cloud Computing. In *2013 IEEE 5th International Conference on Cloud Computing Technology and Science* (pp. 317–320). IEEE. <https://doi.org/10.1109/CloudCom.2013.155>
- [17] Smart Home System using Raspberry Pi. (n.d.). Retrieved from www.ijcsit.com
- [18] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1–7. <https://doi.org/10.1016/j.icte.2017.12.005>
- [19] Delaney, Declan & O'Hare, Gregory. (2014). Modelling the Performance of a WSN with Regard to the Physical Features Exhibited by the Network. 312-324. [10.1007/978-3-319-11692-1_27](https://doi.org/10.1007/978-3-319-11692-1_27)
- [20] Levis, P.; Lee, N.; Welsh, M.; Culler, D. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 5–7 November 2003; pp. 126–137.
- [21] Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Woo, A.; Gay, D.; Hill, J.; Welsh, M.; Brewer, E.; Culler, D. TinyOS: An operating system for sensor networks. In *Ambient Intelligence*; Springer: Berlin/Heidelberg, Germany, 2004
- [22] Morin, Elodie & Maman, Mickael & Guizzetti, Roberto & Duda, A. (2017). Comparison of the Device Lifetime in Wireless Networks for the Internet of Things. *IEEE Access*. PP. 1-1. [10.1109/ACCESS.2017.2688279](https://doi.org/10.1109/ACCESS.2017.2688279).
- [23] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. Pister, "A realistic energy consumption model for TSCH networks," *IEEE Sensors J.*, vol. 14, no. 2, pp. 482-489, Feb. 2014.
- [24] Baines, T., Ziaee Bigdeli, A., Bustinza, O. F., Shi, V. G., Baldwin, J., & Ridgway, K. (2017). Servitization: revisiting the state-of-the-art and research priorities. *International Journal of Operations and Production Management*, 37(2), 256–278. <https://doi.org/10.1108/IJOPM-06-2015-0312>.
- [25] Martín-Peña, M. L., Sánchez-López, J. M., & Díaz-Garrido, E. (2019). Servitization and digitalization in manufacturing: the influence on firm performance. *Journal of Business and Industrial Marketing*, 35(3), 564–574. <https://doi.org/10.1108/JBIM-12-2018-0400>.