Smart Sensing Components in Advanced Manufacturing Systems

Rui Pinto*, João Reis*, Ricardo Silva*, Michael Peschl† and Gil Gonçalves*
*Department of Informatics, Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
{rpinto, jpcreis, rps, gil}@fe.up.pt
†Harms & Wende GmbH
Grossmoorkehre 9, Hamburg, Germany
michael.peschl@hwh-karlsruhe.de

***Abstract*—The latest trends in Intelligent Manufacturing are related with shop-floor equipment virtualization, fostering the easy access to machine information, collaboration among shop-floor equipment and task execution on demand, paving the way for Flexible Manufacturing Systems. Therefore, it allows a high responsiveness to market changes and enables mixed model production. This concept was explored and further developed within an European project called Intelligent Reconfigurable Machines for Smart Plug&Produce Production (I-RAMP^3). The goal of I-RAMP^3 was to contribute to the improvement of European industry competitiveness, by shortening the ramp-up phase times and providing better tools to manage the scheduled and unscheduled maintenance phases. The main step forward on industrial systems was the development of the agent like concept named NETwork-enabled DEVice (NETDEV), which acts as a technological shell to all industrial equipment, both new and legacy. The present paper describes the NETDEV as a whole, applicable to a variety of contexts, but in particular to the virtualization of industrial Wireless Sensor Networks. This virtualization is named Sensor & Actuator NETDEV and extends the current sensor capabilities toward Smart Sensing, allowing for dynamic sensor location, collaboration, diagnostics and reconfiguration. As a technological background, the PlugThings Framework was used for rapid sensor integration of multiple manufacturers, along with UPnP as an enabler for standardized communication and device discovery in the network. The paper concludes by introducing future steps regarding the standardization of the I-RAMP^3 technology.**

Keywords–Smart Components; Wireless Sensor Networks; Intelligent Manufacturing Systems; Industry 4.0; I-RAMP^3.

I. Introduction

I-RAMP^3 is an European Project funded by the Seventh Framework Programme of the European Commission. This collaborative project involves both academic and industrial partners from Germany, Portugal, Netherlands, Hungary, France, and Greece. The vision of the project is to improve the European Industry competitiveness by developing technologies for smart manufacturing systems. To achieve it, the goal is to reduce the ramp-up phase of the shop-floor equipment and manage efficiently the scheduled and unscheduled maintenance phases, increasing at the same time the efficiency of the manufacturing process. By virtualizing all shop-floor equipment into an agent-like system, standardized communication skills and a layer of intelligence for collaboration between, e.g., machines and sensors are introduced, improving also the plug and produce concept towards flexible smart factories. In this context, each agent is represented as a NETDEV, which can represent both physical and logical devices in the shop-floor.

Physical devices deployed on the shop-floor can be both machines - such as a Robotic Arms or Linear Axis - handling systems - manipulators or gantries - buffers, sensors and other actuators. Specifically, sensors have the intent of monitoring the machines' conditions and the corresponding surrounding environment. In contrast, logical devices are virtual instances, which can be responsible for monitoring and diagnosing equipment condition, analysing the production flow or parameter optimization. NETDEVs have a standardized way to communicate using the Device Integration Language (DIL), which is a lightweight and task-driven XML-based language created in I-RAMP^3, in order to ease the quick delivery and reception of process information between all the virtualized shop-floor equipment. The transparency of discovering devices in the network and data exchange between them, using publish-subscribe services, is possible due to *UPnP* as a base technology.

Sensor data is extremely important to monitor machines at the shop-floor level and its environmental surrounding conditions for condition-based monitoring, machine diagnosis and process adaptation to new requirements. The I-RAMP^3 technology allows Wireless Sensor Networks (WSNs) to become more flexible and agile, acquiring new capabilities that can enhance shop-floor operations [1], [2], such as sensor collaboration, which aims for providing to the machine aggregated information instead of quantitative data, and sensor diagnosis and reconfiguration, which aims for detecting sensor malfunctions and correct them without jeopardizing the manufacturing process. Additionally, it allows for dynamic sensor node location used for sensor collaborations, to detect if sensor nodes are physically nearby other sensors and machines, and therefore data can be aggregated for process adaptation and ultimately use of proper instrumentation.

The present work is the result of integrating the solutions reported in [1] and [2] in a standalone technology and applying it in real industrial environment where different case scenarios were explored, not only for verification and validation purposes, but also to assess the usefulness of such approaches. Therefore, a more consistent solution is presented, where the sensor technology for WSN virtualization is integrated with other I-RAMP^3 compliant systems, such as welding machines and vision systems as NETDEVs. This ultimately results in an holistic perspective of the I-RAMP^3 advances in industry, not focusing solemnly on the main functionalities and capabilities, but also on the industrial dynamics of collaboration and the impact of using such a technology.

At the present stage, and based on the advances on WSN communication technologies such as *ZigBee*, 802.15.4 standards [3] and more reliable and long-lasting hardware, in the past few years WSNs became a hot topic for exploration and application in several domains. This is mainly due to its feasibility of installation, when it is difficult to use wired solutions, either by harsh location or high number of sensors used, and due to the easiness of maintenance and reduced costs of cabling [4]. Chen et al. [5] refer as advantages of WSNs their large coverage area, fast communication via Radio Frequency (RF), distributive organisation throughout a direct communication between entities and ubiquitous information. As Ruiz-Garcia et al. [6] pinpoint, some of the WSN advantages can be seen in concrete structures or in the transportation sector, where a controlled environment needs to be monitored in real-time. Additionally, Evans [7] presents enablers and challenges, along with some contextual applicability of WSN in a manufacturing environment and Gungor [8] presents challenges, design principles and technical approaches for industrial WSNs.

Specifically for the industrial domain, Ramamurthy et al. [4] developed a Smart Sensor Platform that applies the plug and play concept by means of hardware interface, payload, communication between sensors and actuators, and ultimately allows for software update using over-the-air programming (*OTAP*). Cao [9] explored a distributed approach to put closer sensors and actuators in a collaborative environment using WSNs. Chen et al. [5] push this approach forward considering the same approach, but taking into account all the industrial domain restrictions like real-timeliness, functional safety, security, energy efficiency, and so forth. All these industrial restrictions and an overview about the industrial domain was explored and presented by Neumann [10]. In the recent past, Chen et al. [11] tackled the Optimal Controller Location (OCL) in the context of industrial environment.

This paper is organized in seven more sections covering all the details about the present work. In Section II, the related work is revised, where several contributions regarding smart production systems are identified. In Section III, an overall description about NETDEVs is made, detailing the NETDEV classification, architecture and communication interface. Section IV depicts the sensor integration on industry and it's virtualization using the PlugThings Framework, detailing and all the capabilities associated with Sensor & Actuator (S&A) NETDEVs, such as collaboration, localization, diagnosis and reconfigurability. Section V talks about different industrial case scenarios used to validate a sensor implementation using the I-RAMP[3] technology, which serves as a proof of concept. In Section VI a discussion about the system and all the functionalities developed is made, Section VII talks about strategies for the future and the importance of standardization, and finally in Section VIII some conclusions are presented.

II. Related Work

In existing production environments, the 'smart factory' concept is still in its early stages. Commissioning is mainly a manual process, where machine parameters have to be found by the operator in a trial and error manner, sensors have to be regularly calibrated and communication between devices has to be established. This process is sometimes supported by software tools and discrete event simulation. This manual process still continues after commissioning, when re-adjustment and reconfiguration of the system needs to be made so the whole production line runs smoothly and efficiently. The same holds true when an industrial facility needs modifications or a production equipment has to be replaced. According to Barbosa [12], traditional manufacturing control systems focuses the processing of a shop-floor control in one central node, which is insufficient to meet current manufacturing requirements that demand flexibility, robustness, reconfigurability and responsiveness. Paradigms supported by decentralization and distribution of processing power are best suited to industrial requirements and constitute in principle a solution towards smart factories. Examples of such paradigms are Reconfigurable Manufacturing Systems (RMS) [13], Multi-agent Systems (MAS) [14] and Holonic Manufacturing Systems (HMS) [15].

Several approaches [16] based on these concepts were developed to support the manufacturing systems complexity, including real implementations in industry. These agent-based applications focused on flexible, reconfigurable and adaptive production at different levels and with different purposes, such as supply chain planning, business process management, production planning, scheduling and optimization, agile manufacturing, enterprise integration, warehouse planning and resource allocation. A very well known approach regarding HMS is the Product Resource Order Staff Architecture (PROSA) [17], which uses holons to represent products, resources, orders and logical activities. PROSA inspired many other approaches latter on, such as a control architecture for an AGV system [18] and an architecture for production control of semiconductor wafer fabrication facilities [19]. In fact, a real application of PROSA was conducted in a packaging cell for Gillete [20], by forming a collaboration between order and resources holons to accommodate changing demands. A more recent architecture called Adaptive Holonic Control Architecture (ADACOR) for distributed manufacturing systems [21] addresses the reaction to emergence and change in environments where frequent disturbances occur.

The industrial acceptance regarding HMS and MAS applications is still low, as pinpointed by Mcfarlane [22], mainly due to the lack of real proof about the applicability of these technologies on real scenarios, aspects related with the technology development process and consequence of the companies' business strategies. DaimlerChrysler applied successfully MAS concepts for both dynamic and flexible transportation and control systems on their production lines [23]. The prototype operated everyday for five years and resulted in a estimated 20% increase in productivity on average. DaimlerChrysler also co-operated with Schneider Electric GmbH to develop a control system for heterogeneous devices in environments with real-time constrains [24]. The US Navy incorporated in their ships a agent-based control system for the heating, ventilation and air-conditions systems [25]. Shen [26] presented a very nice compilation of the main agent-based projects and their achievements.

Before I-RAMP[3], there have been a couple of large projects to set up and improve the framework of HMS, namely the XPRESS project [27], GNOSIS project [28] and PABADIS project [29]. The most relevant aspect to XPRESS was the effort to set up a Scalable Flexible Manufacturing (SFM) architecture, a framework for organizing resources of hardware (machine tools, robots, etc.) and software (cell controllers, process planning, etc.) in computer automated environments, with an emphasis on autonomous decentralized scheduling.

In this approach, each unit in a factory was autonomous and manufacturing execution was the result of negotiations between the autonomous modules with a central 'black-board', containing order information and planning status information. Each resource makes a bid for the work and the best bid wins, leading to an autonomous distributed control.

The GNOSIS project concentrated on configuration systems for design and manufacturing. One part of GNOSIS dealt with 'soft products' and knowledge intensive engineering. In relation to XPRESS, a virtual factory was proposed, which provides reactivity and efficiency by the optimal use of distributed manufacturing resources. These resources are connected to form virtual manufacturing processes, which can be configured and operated as work cells based on product, process or production line principles according to changing demands from the market. The core idea is to have models that communicate with each other, providing both planning and coordination throughout the virtual factories. These GNOSIS concepts have been partly adopted by commercially available planning software. PABADIS demonstrated the advantages of mobile agents compared to classical Manufacturing Execution System (MES) and Supervisory Control and Acquisition (SCADA). Concerning the field level, only fundamental concepts were postulated. However, a flexible production is only possible if the integration of production units at the lowest level (machines, sensors, etc.) is taken into account. This was resolved by the XPRESS project and later, on I-RAMP[3].

In order to make the equipment more versatile, adaptive approaches to encapsulate process knowledge in agent-based production equipment are necessary. I-RAMP[3] incorporated this approach and extended it into a task-based production, where process equipment has expertise about a certain process domain and can execute any task of its domain, based on the description of the task, and can produce a quality result document. In I-RAMP[3], a framework to wrap existing equipment with a NETDEV shell was developed, which contains the required process intelligence and communication means, possible via the exchange of Device Integration Language (DIL) documents.

III. Network-Enabled Devices

NETDEVs are intelligent agent-based production devices that are responsible to equip the conventional manufacturing equipment - both complex machines, as industrial PCs or PLCs, and sensors & actuators - with standardized communication skills, along with intelligent functionalities for inter-device negotiation and process optimization. By wrapping equipment components with the NETDEV shell, they become equipped with built-in intelligence. This means that the NETDEV can incorporate an extensive set of internal models, which are executed on the NETDEV engine. The inherent functionalities include provision of a device self-description, conduction of conditioning monitoring and the provision of a device history, ability to interpret and execute tasks, ability to analyse and optimize a process and ability to perform additional analysis based on knowledge about itself. The NETDEV family is represented in Figure 1.

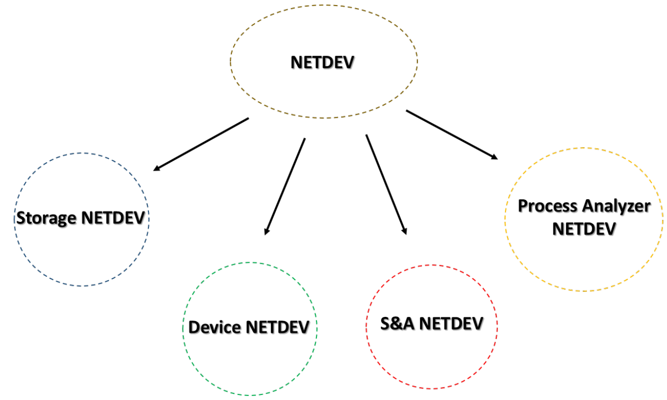

Figure 1. NETDEV Family.

A. *NETDEV Classification*

NETDEVs represent devices that can be categorized into logical and physical devices. Logical devices provide services for data transformation, storage, consumption or production. Physical devices provide physical object transformations or sensing. Therefore, physical devices will, in most of the cases, involve one or more physical objects. A partial NETDEV classification diagram is represented in Figure 2.

Among logical devices, Process and Data Storage can be found, which correspond respectively to Process Analyzer NETDEVs and Storage NETDEVs virtualizations. Process devices can do transformations on the production conditions, for instance the analysis (Analyzer) of production flow, scheduling (Scheduler) or optimization (Optimizer) of NETDEV parameters. The Analyzer provides production analysis of any kind, such as production flow, route optimization, etc. Scheduler correspond to devices that provide production scheduling services to the network, and consume data to feed the required calculations. The Optimizer provides external optimization services to NETDEVs that do not have built-in optimization modules for production parameters. The Process Analyzer NETDEV should be able to perform production parameter optimization for more than one NETDEV, assuming they are from the same type. Data storage devices provide storage means for NETDEVs without built-in storage facilities. NETDEVs should be able to find available Storage NETDEVs on the network and query the free space available, among other properties, to decide where a blob of data should be stored.

As seen in Figure 2, devices dealing with physical objects fit in categories for sensing, transportation or transformation purposes. Sensors correspond to devices that measure some physical property of an object or environment. Actuators correspond to smaller devices (mechanism) that performs some transformation on an object. Both Sensor and Actuators are virtualized into S&A NETDEVs. Stock/Buffer corresponds to devices that provides storage for physical objects. Moreover, Manufacturing corresponds to any device that performs some kind of physical property transformation to another object, such as weld and press. Finally, Handling corresponds to any device that performs some kind of spatial transformation (movement, orientation change, etc.) or holding of objects. Device NETDEVs are virtualizations of Manufacturing and Handling devices.

B. *NETDEV Template*

A model of a NETDEV and its components was developed in order to meet the industrial requirements. According to the

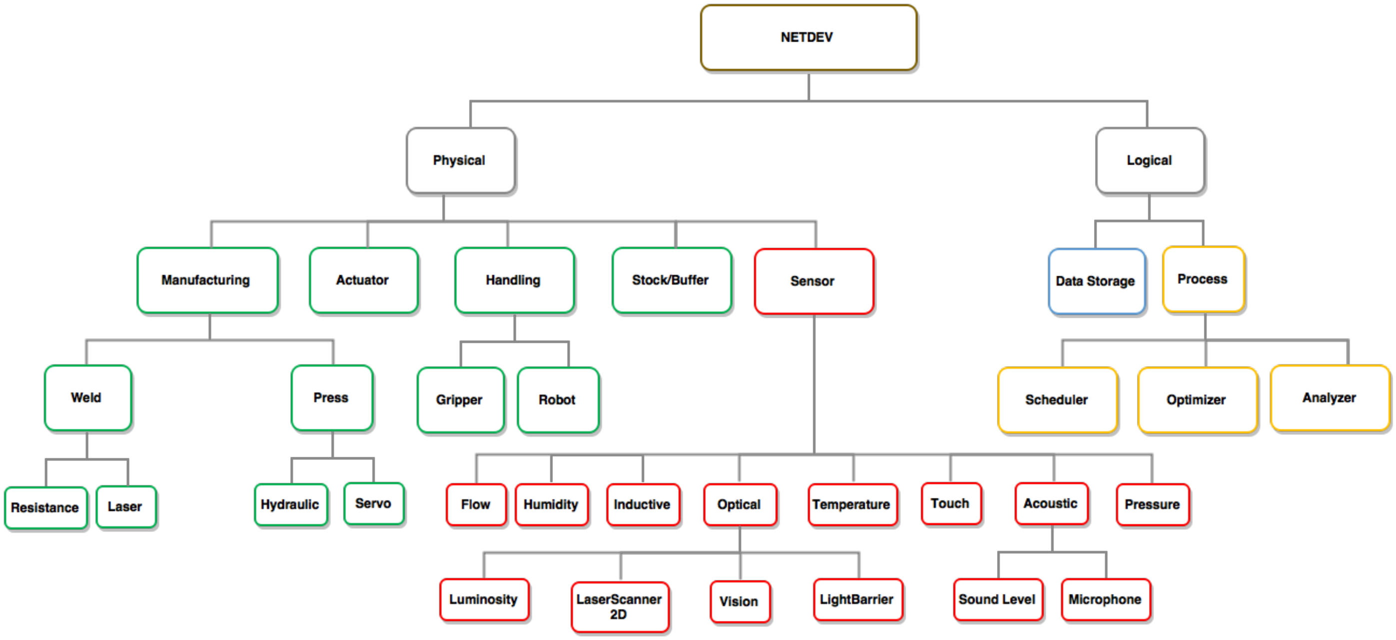

Figure 2. NETDEV Classification.

general I-RAMP[3] approach and architecture, several industrial requirements were derived: reduction of setup time and efforts during re-configuration; reliability of system operations; flexibility in component handling; information provision on device operation; capabilities and procedures classification; interfaces to enable the communication to all other equipment within the architecture; accessibility from "outside" in order to perform maintenance activities; flexible components in order to adapt for new products and variants; possibility to integrate new components; ability of switching tasks within minimal time; and adaptability to different processes and devices

This model was built in the form of a framework, which is used for the implementation of NETDEVs for specific processes. The template comprises both a collection of specifications for NETDEV implementation and a collection of software modules. The user can implement a NETDEV device in three different programming languages, because the NETDEV template is written in *C#*, *C/C++* and *Java*.

Based on the communication requirements, the template includes *UPnP* modules that allow the NETDEV discovery and service publishing. Since the *UPnP* is composed of tow main instances, UPnP Device (information provider) and a UPnP Control Point (information receiver), the developed template contemplates both instances. The UPnP Device is composed by a set of state variables and methods to access the information stored in those state variables. It includes the communication protocol implementation, namely the DIL, the main shop-floor capabilities of the device (represented by the corresponding state variables and methods) and an alive mechanism acting as a "ping" to know if the NETDEV is still in operation. The UPnP Control Point is responsible for recognizing other NETDEVs in the network and subscribe the information generated by the other components, by means of state variables and methods.

C. NETDEV Architecture

Based on the requirements of the developer and/or the equipment that it is intended to be virtualized, the NETDEV template can be used to implement several different types of NETDEVs. On the I-RAMP[3] project, four NETDEVs were implemented, according to the proposed industry requirements. These NETDEVs virtualize the main shop-floor components, both physical and logical. The NETDEV architecture is described in Figure 3.

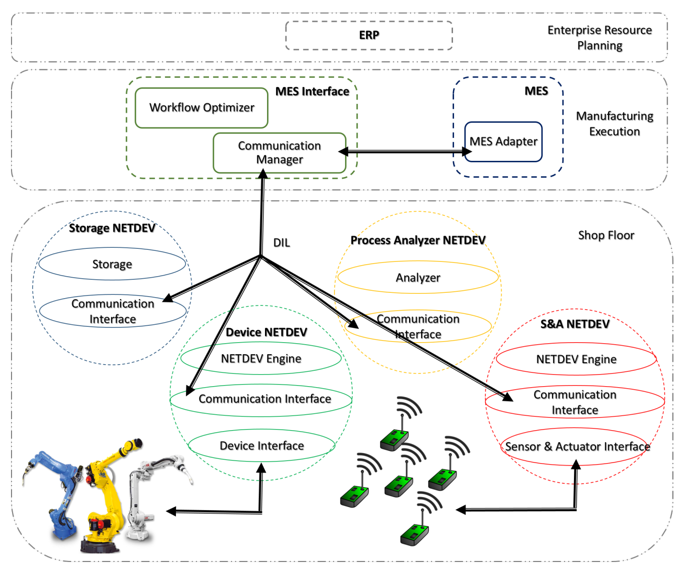

Figure 3. NETDEV Architecture.

The Device NETDEV is a virtual entity that represents any machine or machine component present on the shop-floor, such as manufacturing and handling devices.

The S&A NETDEV is a virtualization of every sensor and actuator present on the WSN used on the shop-floor. Typical used sensors are the Liquid Flow, Luminosity and Temperature sensors. S&A NETDEVs have capabilities such as self-organization, when complex tasks require cooperation between sensors, self-awareness to automatically locate a sensor node, self-diagnosis for sensor malfunction detection and self-healing for automatic sensor reconfiguration.

The Storage NETDEV is a network storage that can either be implemented on a a single device or in separated software products. The goal is to easily extend a storage unit when a new device is plugged in the network. Different types of data can be stored, namely configuration values, counter values and firmware/program files. The system is resilient because one backup file can exist multiple times in the network, distributed over different storage NETDEVs, leading to data redundancy. Also, incremental backups are possible.

The Process Analyzer NETDEV is used for a production process analysis, such as welding or polishing of an equipment for condition monitoring, including sensor & actuators. This NETDEV allows the visualization of relevant Key Performance Indicators (KPI's). The analysis is dedicated to the condition monitoring and visualization of, first, a welding process quality, second, welding timer monitoring for equipment breakdown and, third, sensor group monitoring for sensor breakdown.

D. NETDEV Interfaces

NETDEVs are able to describe and optimize themselves towards their environment by providing knowledge and models about their properties, abilities, constraints and reuse abilities. Furthermore, they have the ability to perform condition monitoring and maintain a device history, interpret and execute tasks, optimize process, expose abilities and to predict its maintenance requirements.

A NETDEV has two main interfaces: the communication interface and the device interface. The communication interface handles all data exchange among NETDEVs and the Manufacturing Executing System (MES). It comprises the exchange of documents described in the DIL. The device interface is used to link physical devices and tools to the NETDEV. The NETDEVs are able to work with different tools and devices that have similar features, adapting themselves to the discovered unit.

The NETDEV communication is divided in three main steps. First, when the NETDEV enters the network it announces itself and gets knowledge about the existing NETDEVs already in the network. This discovery process of entities on the network is automatically done by the *UPnP* technology. Second, DIL is used exclusively for NETDEV communication, when tasks should be requested. It basically consists of document exchange between NETDEVs. Third, task content and data is specified and exchanged within the DIL documents. A simple case of NETDEV communication using DIL is presented in Figure 4.

DIL implements four different types of Extensible Markup Language *(XML)* documents and each one can be exchanged inside the environment between the NETDEVs. The four types are: NETDEV Self-Description (NSD); Task Description Document (TDD); Task Fulfilment Document (TFD); Quality Result Document (QRD).

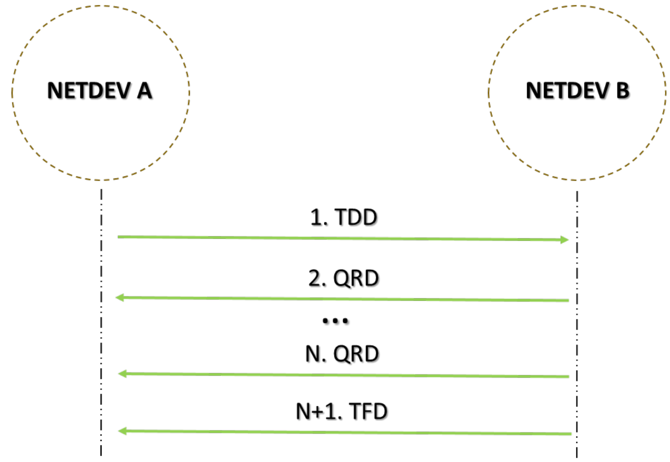

Figure 4. DIL Communication.

The NSD describes the capabilities of a NETDEV, which is basically a range of tasks that can be performed by the NETDEV. The tasks may be defined as goals and conditions, or as bare process parameter values. The task range gives the possible range of goals and conditions or parameter values, which can be realized and accepted by the NETDEV according to its physical capabilities. Additionally, NSD can be adapted, when self-diagnosis finds process restrictions.

The TDD describes the information defining a requested task as roughly specified on NSD. It determines one of the possible goals or parameter values, which have to be reached by the NETDEV. If it is a continuous task (for instance, detection of irregular signal values) or if it is a repetitive task, the period of the task execution or the number of task repetitions is given.

The TFD is a document-type acknowledge to the TDD, reflecting the settings with respect to the requested goals or parameters. The TFD also has a second purpose: It is used to inform other NETDEVs about the actual settings and let them decide if they can cooperate with the NETDEV under the present circumstances or if they have to wait until they can set them otherwise, via a new TDD.

The QRD describes the result achieved after process execution, which can be the description of the end state or the quality achieved after the process execution. In a continuous or repetitive process, the QRD is issued only at the end of all scheduled repetitions or time period and is giving a summary of the total repetitive or continuous process.

IV. Sensor & Actuator NETDEV

Sensor usage on industrial applications has become extremely important, since monitoring the behavior of a machine is crucial to adapt its operation due to regular changes on product demand. On a shop-floor environment, sensors should not be treated as an integral part of a machine, but a separated component, which like complex machines, should be flexible enough to change its operations according to process demands. In I-RAMP[3] were explored new concepts on WSNs applied in industry, aiming for the addition of an intelligence layer on sensors, which empower them to be as complex as machines, both sharing plug and produce features and both capable of communicating with each other on an agent-like system environment.

Intelligent WSNs rely on some features such as easy integration of sensor nodes from different manufacturers using, e.g., the PlugThings Framework [30], along with automatic calculation of the nodes' physical location, self-diagnosis capabilities using sensor data validation methods, and self-reconfiguration capabilities using *OTAP* technologies to re-program new sensor nodes on the network.

A. Sensor Integration

In the I-RAMP[3] project, the integration of multiple types of sensor nodes on the system is made using the PlugThings Framework [30], which contains a Universal Gateway (UG) to parse raw sensor data from the different sensor nodes. As can be seen in Figure 5, each sensor node of the network communicates directly to this gateway node, where the received measurements are processed and translated from raw data (stream of bytes) into readable form (measurement values). These data are compiled on *XML* based format files that are part of the Sensor & Actuator Abstraction Language (SAAL), which is used to communicate with Sensor & Actuator Abstraction Middleware (SAAM). All the intelligence related to the sensors is implemented in SAAM. When the SAAM receives a new message from a sensor node, it will collect the sensor board identification number (ID) and the communication module Media Access Control (MAC) Address. Both board ID and MAC Address are the unique identifier of a sensor node.

Joining a new sensor node to the network will imply the creation of a new S&A NETDEV corresponding to that sensor node, letting transparent to all the entities on the network what measuring tasks it can perform. Since a sensor node can have multiple sensors integrated, the corresponding S&A NETDEV will be able to perform different tasks related with the different sensor types of the sensor node. It will have one task per sensor integrated in the mote, being this way able to provide sensor information in a standardized way.

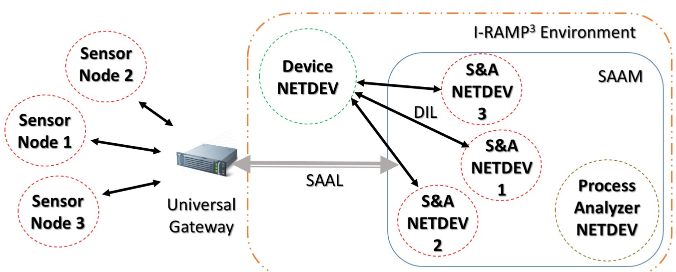

Figure 5. I-RAMP[3] Environment.

Basically, S&A NETDEVs have one functionality, namely task execution, to provide sensor information to other entities per integrated physical sensor in the sensor node. S&A NETDEVs can easily communicate with other NETDEVs on the network using DIL, such as Device NETDEVs and Process Analyzer NETDEV, which monitors sensor behavior while in a group collaboration. At this stage, S&A NETDEVs can execute two different tasks, both usually requested by a Device NETDEV: *Measurement* and *Group Formation* tasks.

1) Measurement task: Is used when the Device NETDEV needs the measurements of a single sensor node. Therefore, it should specify the desired type of sensor to receive the corresponding sensory data, the frequency of the readings, sensor accuracy, coverage radius of the sensor in spatial units (if applicable) and the number of cycles to execute the task.

2) Group Formation task: is requested when the Device NETDEV aims to collect several measurements at different locations, which means having multiple sensors executing the same task at the same time. In this specific task, the S&A NETDEV that receives the task is responsible for choosing possible S&A NETDEVs candidates to join the group, based on the task parameterization and the sensor location. This allows for a more distributed approach in terms of collaboration, rather than a peer-to-peer-like solution, implying a communication with all the S&A NETDEVs from a group instead of only one. In terms of parameterization, beside the desired type of sensor to receive the specific data, frequency of measurements, sensor accuracy and the number of cycles to perform the task, the *Group Formation* parameterization must also specify the number of sensors intended in the group.

B. S&A NETDEV Collaboration

S&A NETDEV *Group Formation* is a methodology used to improve the communication performance and reduce complexity between Device NETDEVs and S&A NETDEVs while executing tasks with a sensor collaboration nature. Thousands of sensors can exist on the shop-floor level, and therefore, the flow of information can be very high when requesting tasks. The *Group Formation* methodology is a more distributed approach that allows S&A NETDEVs to provide a more aggregated information when the task requested from a Device NETDEV requires measurements from more than one sensor node. Instead of establishing communication with every S&A NETDEV required, the Device NETDEV will have a single point of communication with one S&A NETDEV, which is responsible for forming and managing a S&A NETDEVs group.

The main premise for the *Group Formation* is that every S&A NETDEV is capable of forming a group. When a Device NETDEV requests a S&A NETDEV to form a group with a certain number of sensors, this S&A NETDEV is responsible for searching in the network, communicating via DIL, for available S&A NETDEVs that are capable of performing the same task as it and the corresponding sensor nodes must be physically located in the same production area. If the number of group members reaches the requested number of S&A NETDEVs in the beginning, the S&A NETDEV responsible to form it becomes the group leader, called Super S&A NETDEV, and the group is formed. Internally in the group, each S&A NETDEV will collect measurements during the requested number of cycles and the Super S&A NETDEV is responsible, not only to gather all sensor data, but also process it to a more meaningful value, to be sent afterwards to the Device NETDEV. When task execution ends, the Super S&A NETDEV will terminate the communication with the Device NETDEV and release the S&A NETDEVs from the group, which become available to execute other task requests from other NETDEVs. Figure 6 compares both peer-to-peer and sensor collaboration approaches.

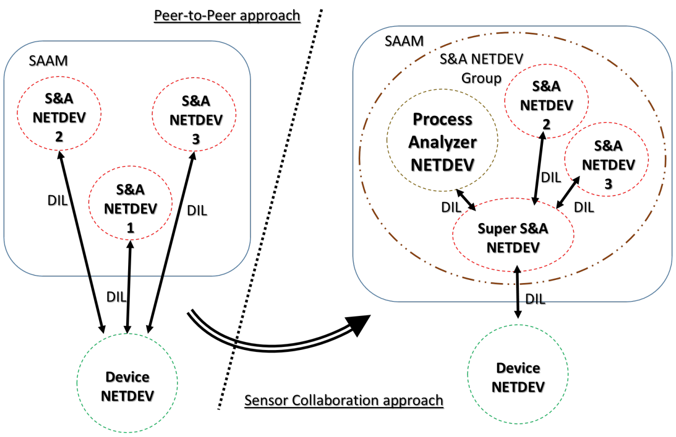

Figure 6. *Group Formation* Schema.

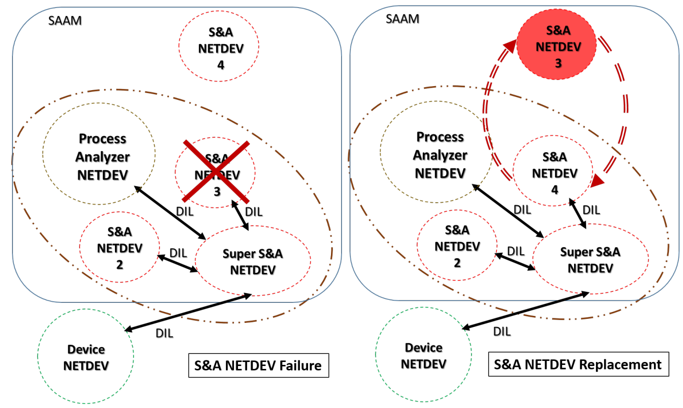

Figure 7. *Group Formation* - S&A NETDEV Failure.

An additional NETDEV entity represented in Figure 6 is the Process Analyzer NETDEV, which is created by the Super S&A NETDEV when the group is formed. This virtual entity is responsible for applying sensor validation techniques, such as the Spatial Correlation technique [31], [32], to assess the condition of the group based on the sensor data generated. The Process Analyzer NETDEV collects the sensor data from each element of the group and identifies the most devious data set by comparing the data sets from all group members. If the deviation is greater than a predefined threshold, then the sensor node is classified as probably malfunctioning. The Process Analyzer NETDEV reports to the Super S&A NETDEV, via DIL, the existing of a malfunctioning group member at that time, so it can make a decision about the faulty sensor node(s) and maintain the group functionality as consistent and reliable as possible.

With the *Group Formation* task, there are two main benefits from the task requester perspective. Assuming a Device NETDEV wants to collect and analyze data from multiple S&A NETDEVs: first, it avoids communicating with several S&A NETDEVs at the same time to collect data, since the responsibility to form a group is on the S&A NETDEV; second, the S&A NETDEVs can process all sensor data and provide a statistical description, passing the data analysis complexity to the group side. This means that the requester does not need to know any statistical technique to process the data from multiple sensor entities on the network. However, relying on one single point of communication, increases the vulnerability in case the task execution fails on that point. Hence, there are two failing scenarios on a group: 1) Failure of the Super S&A NETDEV or 2) one or more S&A NETDEVs from the group fail(s).

1) Failure of the Super S&A NETDEV: If the Super S&A NETDEV fails, the single point of communication supporting the interaction between the Device NETDEV and S&A NETDEVs from the group is lost. There will be no more conditions to continue with the task execution, so the task stops and the group is disaggregated. In the termination process, the Super S&A NETDEV is responsible for changing the process state of the remaining group members, so they can stop executing the *Measurement* tasks for the group, becoming available to perform new tasks upon request from other NETDEVs.

2) Failure of one or more S&A NETDEV(s) of the group: If a S&A NETDEV from the group is failing, the Super S&A NETDEV is still working correctly, so the group is not in danger of collapsing and the communication with the Device NETDEV is not affected. In this case, the Super S&A NETDEV is responsible for replacing the failing S&A NETDEV for a new one able to join in. While the replacement process occurs, the collected data from the group will be less accurate, because the results sent to the Device NETDEV do not contemplate all the requested NETDEVs, due to a temporary deficit of one S&A NETDEV (the malfunctioning one). Figure 7 depicts the process when a S&A NETDEV (in this case, S&A NETDEV 3) fails and is replaced by an available S&A NETDEV (in this case, S&A NETDEV 4).

C. S&A NETDEV Location

WSNs applied in industry are used to monitor different production cells on the shop-floor, consisting of spatially distributed sensor nodes, which are equipped with several sensors to monitor the environmental conditions surrounding the cells where they are located. If a machine, located in one of the production cells needs information about, e.g., the luminosity conditions surrounding the cell to execute a given task, the machine may require, from available sensor nodes placed in that production cell, valuable information for process parameterization.

In the I-RAMP[3] context, the Device NETDEV that is requesting the task should search on the network for available S&A NETDEVs with the required capabilities (described in the NSD), e.g., measuring luminosity conditions and, consequently can form a sensor group that measures luminosity. Facing a request for a *Group Formation* task from a Device NETDEV, the S&A NETDEV will only accept the task if the corresponding sensor node can fulfill the required parametrization and is physically located on the same area as the machine that requested the task in the first place.

Every NETDEV is characterized by its task execution capabilities (NSD) and the area on the shop-floor where the correspondent equipment is located. On contrary of machinery, the location of sensor nodes can be calculated dynamically by a S&A NETDEV location system, which uses the incoming RF signal strength of the sensor node on several beacons for position estimation. Beacons are physical entities located in known strategic positions of the shop-floor, mainly in the limits of shop-floor sections like cells or production lines and are responsible by receiving messages from sensor nodes, assess their signal strength and position in order to assign the current relative location to S&A NETDEVs. At this point in the implementations, only sensor nodes that are using *XBee* radio modules are acceptable to calculate dynamically the location.

Location systems on WSN is a very active research area and there is no universal solution for this topic. The main goal is to identify the physical location of a sensor node on the WSN. Each approach of node location is fitted to a specific operating environment, such as indoors or outdoors spaces like urban areas, forests or even underwater. In the industrial context, estimating the node positions in meters is not important, as the main goal is to find in which section on the shop floor the sensor nodes are located. The algorithms for node location are made of two main components: 1) Estimation of distance or angle between two nodes and 2) Calculation of the node position. First, the distance or angle between two nodes must be estimated to be used on the calculation of the node position related to one or more anchor nodes (nodes with a previously known location - beacon). Then, the information about the distance and the position is used by an algorithm to determine the node's location.

There are several methods [8] to estimate the distance or angle between two sensor nodes. Some are more precise than others, but on the other hand, they require more hardware resources, consume more energy and demand more computational power. Time delay based methods, such as Time Of Flight (TOF) [33], estimate the distance by measuring the time it takes for the RF signal to travel between them. Since the RF signal travels at speed of light, it could be extremely difficult to measure the signal time travel. So, time difference methods, such as Time Difference of Arrival (TDOA) [34], measure the difference on the time travel between the RF signal and an acoustic or ultra-sound signal, which, because it travels at the speed of sound, it must be easier to measure. This method requires extra hardware such as transmitters and receivers of ultra-sounds. Signal angle/direction estimation methods, such as the Angle of Arrival (AOA) / Direction of Arrival (DOA) [35], [36] is a method that uses the RF signal angle of arrival to determine the sensor node position. Again, the method requires extra hardware such as specific antennas in both transmitter and receiver. For last, the Received Strength Signal Indicator (RSSI) [34], [37], [38] estimates the distance based on the strength of the RF signal, which are theoretically inversely proportional, if perfect conditions existed. In comparison with the previous methods, this one has the advantage of no extra hardware is required, other then a simple antenna. The disadvantage is the lower precision of measurements when signal noise and interference exist.

In the I-RAMP[3], distance estimation is done between the sensor node and anchor nodes placed in know shop-floor locations, using the RSSI method. The considered propagation model is the Free Space model [39], [40]. Although having a lower precision (2m to 5m errors), the distance estimation is acceptable for the method applicability in the industrial scenario.

The radio signal is highly susceptible to noise [41] caused by reflection, refraction, diffraction, scattering, fading, inter-symbol interference and shadowing. Consequently, there will be distance deviations in the end. This can be minimized by filtering the signal using a moving average to better approximate the path loss logarithmic curve. The path loss coefficient is determined dynamically using path loss log-distance model using measurements of RSSI between beacons, using (1), where $P(d)$ is the RSSI in dBm, $P(d_0)$ is the RSSI at a fixed reference distance from the transmitter d_0, n is the path loss coefficient, X_σ is a normal random variable used to modulate, d is the distance in meters between transmitter and receiver, P_{TX} is the transmission power and A is the signal attenuation. Manipulating the formula, first the path loss coefficient is calculated using (2), where the RSSI and distance are between beacons. Then, (3) is used to calculate the distance between a sensor node and a blind node.

$$P(d) = P_{TX} + A - 10n \times log(\frac{d}{d_0}) + X_\sigma \tag{1}$$

$$n = \frac{|P(d_0) - P(d)|}{10log(d) \times 2} \tag{2}$$

$$d = d_0 \times 10^{\frac{|P(d_0)-P(d)|}{10n}} \tag{3}$$

The node position is calculated using the distance estimation of three anchor nodes closest to the sensor node with the Bounding Box method [38]. Bounding Box is a variation of the trilateration, which uses the position of three anchor nodes, with known positions and distances between them, to calculate the position of the sensor node, as shown in Figure 8.

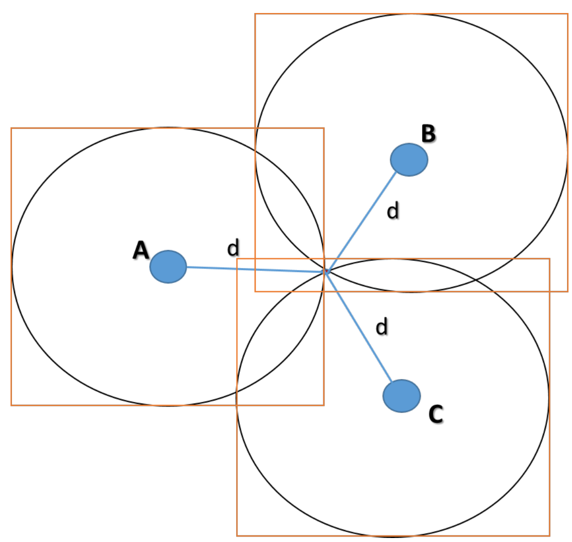

Figure 8. Trilateration and Bounding Box for node position estimation.

The position of the node is calculated by the intersection of three circles, each one is centred on the anchor node and with radius equal to the distance to the unknown position node. With Bounding Box, the calculation complexity is reduced by replacing the circles by squares. The intersection of the different squares results in a rectangle, where the centre is the estimated position of the node.

D. S&A NETDEV Diagnosis

Sensor data is used as an input for complex machines to control the manufacturing process and to adapt themselves according to external conditions. This adaptation allows the machine to be flexible enough to change its variable inputs and internal processing, controlling the production process to maintain product quality despite fluctuations. Machine's process depends on data measured from sensors, so it is very important that data stays the most reliable as possible when delivered to the machine. Data samples collected from sensors, especially from WSNs, are prone to be faulty due to internal and external influences, such as environmental effects, limitations of resources, energy problems, hardware malfunctions, software problems, network issues, among others, as shown in [42]–[45].

Sensor data validation consists of a set of methods applied to the data provided by the sensors with the main goal of detecting anomalies and malfunctions on these sensors and take action accordingly on the corresponding S&A NETDEVs. But, finding deviations from normal sensor readings do not necessarily mean that they occur due to a malfunction of the sensor node. Instead, they can occur due to an abnormal variation of the environmental conditions being measured. Despite being a sensor-based or an environment condition-based cause, each sensor node of the WSN is aware of its state and is capable of performing self-diagnosis during the task execution.

Anomaly detection methods generally classify data into correct or faulty. There is no right method that works better than all the others and no method guarantees success, because they all depend on several factors such as type of monitored variable, the overall measurement conditions, the sensor used and the characteristics of the environment being perceived [46], [47]. In [31], [46] is proven that anomaly detection should not rely on just one method, but instead on a number of methods applied successively for detecting different types of data faults. Furthermore, there are methods [46] suitable to be used online, and other more complex and demanding on the processing level, suitable to be used offline. Offline validation methods, such as Bayesian Networks (BNs), Artificial Neural Networks (ANNs), Regression Techniques like Partial-Least Squares Regression, etc., are used in many different contexts such as aerospace, energy, electric power systems, urban environment, among others [48]–[54]. Regarding S&A NETDEVs, techniques that provide a quick WSN diagnostics were used, such as Min/Max, Flat Line [31], [32], Modified Z-Score [55] and No Value detection.

The Min/Max approach is based on a heuristic rule, which defines upper and lower bounds that refer to hardware specifications or/and conditions that are not likely to occur in the current context. Therefore, if sensed data is within bounds, data are likely good, otherwise, the sensor may be faulty. The Flat Line technique is based on temporal correlation of a data set of the most recent collected measurements. If the difference between successive data samples remains zero, this means that the sensor is probably faulty. Modified Z-Score is a statistical-based technique used as an outlier detection mechanism. It takes into account averaged values and deviations to assess if a certain value do not follows the same behavior as others values in the data set. The No Value detection technique finds gaps in data sets. If the difference between the current time and the time stamp of the last measurement is unusually large, then probably the sensor has stopped the communication with the gateway.

On I-RAMP[3], the sensor data validation is characterized by four main steps, as shown in Figure 9: 1) First, raw data is acquired from the sensor nodes; 2) Raw data is converted into a readable form by the UG and sent to the SAAM; 3) While a S&A NETDEV executes a task, the received sensor data is validated by a sequence of internal methods to detect anomalies; 4) If anomalies on data are detected, the corresponding S&A NETDEV is marked as probably faulty, which results, depending on the severity of the error detected, in the inability of accepting future task requests or termination of the current task's execution.

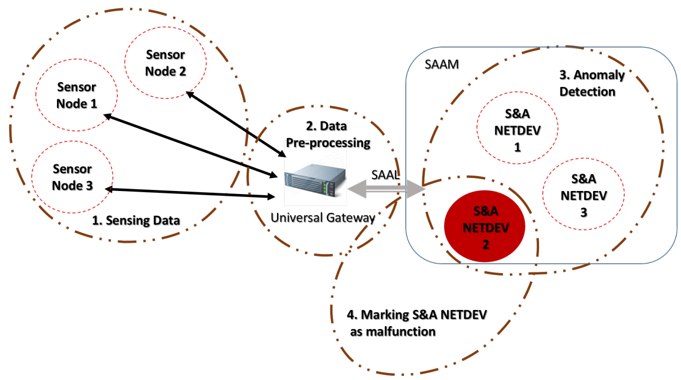

Figure 9. Sensor Data Validation approach on I-RAMP[3].

While the S&A NETDEV is executing a task, the data set of the corresponding sensor node will go through two validation modules: Module A, which is intended for detecting sensor malfunctions and Module B, which is intended for detecting abnormal behavior from the sensor node.

Module A validates the received sensor data using the Flat Line [31], [32], [56] and No Value detection methods, aiming to identify a malfunction sensor node. If the Flat Line method returns positive for error detection, it means that, on the sensor node, the board is reading the same electrical quantity for an unusual amount of time. This implies that the sensor is not detecting any variation on the environment quantity being measured or is failing to do so. Since electrical signals are time varying analog signals of voltage, current or frequency, usually associated with noise, it is very difficult to have sequence samples with no difference between them. Hence, when this occurs, it is most likely that the sensor is not correctly connected to the board. On the other hand, if the No Value method detects gaps in the data set, most likely the battery ran out of energy or the sensor node just broke down. Facing a malfunctioning sensor node, the corresponding S&A NETDEV is responsible for terminating prematurely the task execution, without any human interaction and making itself unavailable to take on other task requests.

Module B is intended for methods that detect outliers, such as the Min/Max detection [31], [32], [57], which detects readings out of system limit thresholds, and the Modified Z-Score [32], [58] that detects spikes and abnormal readings. This module returns a strong probability about the malfunctioning state of the sensor, despite lower than the one returned by Module A. This probability is based on the defective readings that, in this case, can be caused by sensor failing or abnormal behavior of the system itself. In such circumstances, the S&A NETDEV waits for the normal task termination, becoming unavailable regarding the acceptance of future task requests, while a maintenance process does not occur on the corresponding sensor node.

E. S&A NETDEV Reconfiguration

Over the Air Programming (*OTAP*) is a technology developed originally to update firmware for mobile devices. Since the use of this type of equipment greatly relies on wireless Internet access, *OTAP* has been used on the past years, from manufacturers to network operators, to deliver firmware updates to equipment with Internet access. However, because of the wide use of WSNs and their growing complexity, *OTAP* has been taken to a new direction towards WSNs [59]. A WSN could have thousands of sensor nodes and the maintenance of these nodes could be very time-consuming. Therefore, since they must all be re-programmed one by one, this is not a very cost-effective solution. Moreover, the WSN may have nodes located in difficult access places, so updating firmware in sensor nodes on site can be challenging. Several sensor nodes from different manufacturers are already embedded with the *OTAP* technology, which relies on updating firmware on sensor nodes from the gateway node, using the existing wireless communication between them, such as *XBee*, *Wi-Fi* or *3G*.

On I-RAMP^3, the WSN consists in different sensor nodes, gateway nodes connected to the UG and the communication topologies between them. The sensor node used in the I-RAMP^3 to proof of concept is the Libelium Waspmote PRO (v1.2) [60] sensor boards, equipped with the *XBee* radio module for the 802.15.4 communication protocol [3]. Updating firmware on the Waspmote PRO (v1.2) requires using the Libelium *OTAP* technology [6], which divides the OTAP process on two main steps: 1) node discovery on the network and 2) firmware upload. The OTA-Shell application [61] is used at the UG level to control the options available in *OTAP*, sending commands to the sensor nodes to be reprogrammed. A firmware upload occurs when the shop floor operator replaces sensor node hardware due to a severe malfunction detection on a sensor node (using the methods discussed previously). The logical representation of *OTAP* methodology in I-RAMP^3 is represented in Figure 10.

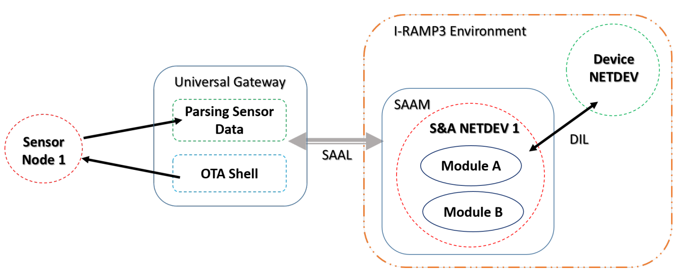

Figure 10. *OTAP* Methodology on I-RAMP^3.

When a S&A NETDEV is executing a task and a sensor node failure is detected, the malfunction could be caused by irreversible problems that require equipment replacement on the nodes, such as: 1) replacement of a bad sensor or communication module; 2) replacement of a bad sensor board; 3) replacement of the entire sensor node. Replace a bad sensor or communication module does not require firmware update of any kind, since these components are external to the sensor board that is running the firmware. On the other hand, when replacing a bad sensor board or the entire sensor node, a firmware update is required, which can be done traditionally or using the *OTAP* approach.

Traditionally, before a new sensor board is connected, it needs previously to be manually flashed with the right firmware. This approach may be counterproductive on a smart factory context, since the ramp-up time of replacing a sensor board could be very high. With the I-RAMP^3 *OTAP* approach, when a new sensor board is connected to the network, the sensor node is flashed automatically with the correct firmware using *OTAP*. The basic idea is to previously store on the UG the replaced sensor node's program in form of an automatic generated binary image after compiling the code and flash the new sensor node over the air with the stored binary image, replacing a malfunctioning one.

1) Faulty Sensor or Communication Module Replacement: Sensor node malfunctions may have its root cause on specific components of the node, leading to the replacement of only the bad component. A malfunctioning S&A NETDEV detected by, e.g., a Flat Line, could be possibly caused by a broken sensor that was used on the task execution requested. Therefore, the replacement process requires only the exchange of that specific sensor. On the other hand, if the malfunction is detected by, e.g., a No Value method, probably it is caused by problems on the communication module. The S&A NETDEV becomes temporarily unavailable, until the component exchange, by a shop-floor operator, is finished. When the replacement of the sensor or communication module is finished and the sensor node is connected again to the network, the UG will detect incoming readings from the same sensor board once again. Then, SAAM associates this new sensor node to the previously unavailable S&A NETDEV, making it available for task execution once again. If the communication module was replaced, the MAC Address associated with the S&A NETDEV is updated by the new one.

2) Sensor Board Replacement: In the I-RAMP^3 context, *OTAP* is used not for firmware update but for flashing a new sensor board for the first time it connects to the network, after replacing a failing sensor node. The process begins the moment a malfunction sensor node is detected during task execution and, a shop-floor operator diagnosis confirms the root cause being on the sensor board. This implies replacing the failing sensor board only, without exchanging good components connected to it, such as sensors and communication module. With the I-RAMP^3 *OTAP* approach, the shop-floor operator avoids flashing manually a new sensor board before it is connected to the system.

From the UG perspective, a sensor node can send two types of messages to a gateway node: sensor reading messages containing the sensor node information, as the actual measurement made, and "Alive" messages containing only information about the sensor node. The sensor node information required are the sensor board ID and MAC Address of the communication module. When the UG detects an "Alive" message, it means that the sender sensor node is not running any firmware yet and is waiting to receive instructions for an *OTAP* process. Because only the sensor board was replaced, the UG detects a MAC Address that was already associated with an existing S&A NETDEV but a different sensor board ID. This means that, despite a new sensor board, the sensor node is associated once again to the previously unavailable S&A NETDEV. Since "Alive" messages are received instead of sensor readings messages and the sensor node is associated with an existing S&A NETDEV (due to the matching MAC Address), a new *OTAP* process begins.

First, SAAM identifies which firmware is the right one to flash the sensor node via *OTAP*, based on the previously created S&A NETDEV capabilities. Hence, SAAM informs the UG to start a new instance of the OTA Shell, using the identified binary image to flash the specific sensor node. The UG runs the OTA-Shell, which starts by scanning the network to locate the new node to be flashed. When the sensor node is found, the UG sends the binary file to the identified node, which stores the file on the Secure Digital (SD) card. After receiving the program successfully, the sensor node reboots several times in order to start the execution of the new firmware. The firmware is copied from the SD card to the Flash Memory and the sensor node starts running the new binary file. After restoring its configuration, the sensor node is ready to operate again, by measuring again the environmental conditions and send the results to the gateway node. The UG will receive again sensor reading messages and the corresponding S&A NETDEV will change its internal state, becoming available for task execution once again.

3) Replace the Entire Sensor Node: The malfunctions detected may be severe to the point where none of the component on the sensor node can be saved, forcing the replacement of the entire sensor node. When this happens, a new sensor node is connected to the network, which will send to gateway node a sensor node ID and MAC Address that are new to the UG, resulting in the creation of a new S&A NETDEV by SAAM, available to take requests for task execution. The only way SAAM knows which tasks the new S&A NETDEV is able to perform, is by parsing the messages received from the sensor node and detect which are the sensor types connected to it. This occurs when the new sensor node is already flashed with the right firmware for the task intended. On the other hand, if "Alive" messages are received, SAAM can not possibly know which tasks the new sensor node is able to perform, because the messages received do not have any sensor readings to be identified and SAAM does not have a background to associate this sensor node to an existing S&A NETDEV with capabilities already identified.

V. I-RAMP^3 TECHNOLOGY VALIDATION

The results generated in I-RAMP^3 are relevant to the European manufacturing industries, specially in the field of fast ramp-up and commissioning of production lines. In order to ensure maximum impact, physical demonstrators were built in order to demonstrate and validate the I-RAMP^3 concept, which are described next. The aim of this section is to identify the demonstration goals of the project, according to the industrial requirements defined earlier, and to describe and discuss the demonstrators results, focusing on the sensor & actuators component implementations. The partners of the I-RAMP^3 consortium elaborated and compiled their current practises and problems into a number of quality goals, based on the results obtained on the I-RAMP^3 approaches. Some of these quality goals are shown on Table I.

TABLE I. I-RAMP^3 consortium current quality goals.

Quality Goals	Current Approach	I-RAMP^3 Approach
Time for component exchange	30 - 90 min	<5 min
Number of manual backup operations	10	0
Timeliness of backup data	Depending on the backup interval (e.g., daily or weekly)	Real-time
Configuring welding parameters	3 days	<2 hours
Adapting a welding process	1 day	<2 hours
Malfunction analysis time	Several hours	A few minutes
Level of setup personnel	High-skilled staff	Technician or novice engineer
Process parameter configuring	Several hours	A few minutes
Adapting to a new product	Several hours	A few minutes
Flexibility in use of standard subsystems	Manually configured	Self configuring subsystems

The demonstrators quality goals are classified into seven categories, namely reduction of process configuration time and efforts, faster and better local and remote trouble shooting capabilities, easy addition of sensors, actuators and production units to the production network, easy integrated management of the production network and proactive maintenance scheduling. Most of these goals were tested and validated in three main demonstrators, namely the demonstration towards 1) set-up and ramp-up of a new E-Vehicle assembly line, demonstration towards 2) enhancing device with re-use and predictive maintenance capabilities and finally the demonstration towards 3) component exchange in E-Vehicle subassembly unit. For the Sensor & Actuators components only the first two demonstrators were validated.

A. Demonstration towards set-up and ramp-up of a new E-Vehicle assembly line

This demonstrator consisted of validating equipment and sensor auto-detection in production environments, flexible and easy-configurable production using task-driven manufacturing and rapid ramp-up after device breakdown or configuration failure. Physically, the demonstrator used a welding machine cell, promoted by AWL. This cell consisted in a safe loading area where the parts to be welded are manually placed in a jig. Once loaded, the jig validates the parts, which are welded by one or more specific welding robots in a specific process (resistance or laser). The assembled product is subsequently manually processed or taken out by a handling robot for the next process step. The welding quality is monitored by a welding process controller. Moreover, the health state of the water cooling system of the cell was performed using liquid flow sensors. Figure 11 represents the AWL welding machine used in this demonstrator.

1) Rapid Ramp-Up After Device Breakdown: This use case aims to validate how a S&A NETDEV can detect malfunctions on the corresponding sensor node and, after replacement, how can the new sensor node learn how to perform the same task as the replaced sensor node did. In the welding cell there are two water flow sensors, which are compliant with the I-RAMP^3.

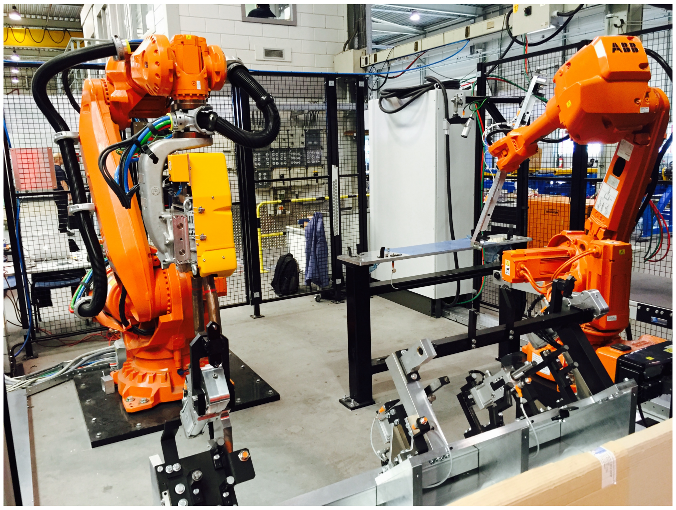

Figure 11. I-RAMP[3] demonstrator regarding set-up and ramp-up of a new E-Vehicle assembly line.

This allows other NETDEVs in the cell to quickly monitor the sensor's readings, unit of measurement, boundaries, etc., without worrying about the hardware specifications. In order to monitor the state of its cooling system, the Device NETDEV, which corresponds to the cell unit, requests a water flow *Measurement* task to the S&A NETDEVs. Figure 12 is a screenshot of the PlugThings Framework, which represents the water flow readings of one flow sensor used in the welding cell.

In this case, the maximum flow rate, expressed in liters per minute (l/min), of the sensors used is around 30 l/min and the minimum is 0 l/min. By the analysis of Figure 12, the normal flow rate of the welding cell is around 9 l/min and the maximum variation detected is +/-0.5 l/min. The flow rate of the welding cell could also be controlled manually, by opening or closing a water tap. One can observe this manual variation around the timestamps 01:54 pm until 01:55 pm. Regarding the perspective of the S&A NETDEV, Figure 13 represents the S&A NETDEV information before (process state is ”Standby”) and after (process state is ”Productive”) the water flow *Measurement* task execution. Besides the S&A NETDEV process state, other information is available, such as the type of the corresponding sensor, the most recent sensor reading, timestamp of each sample, cell or shop-floor area where the sensor node is located and the *OTAP* state.

To validate this use case, a malfunction on the sensor node was simulated, by disconnecting a wire on the sensor board during the *Measurement* task execution of the S&A NETDEV. This simulation results in misleading sensor readings (in this case 0) without any variation what so ever, as can be seen on timestamp 01:58pm in Figure 13. The S&A NETDEV quickly detects a Struck-at-fault type error and stops the task execution and change its own state to ”Engineering”. This process state means that the S&A NETDEV becomes unavailable to continue or accepting new task requests until maintenance is performed on the corresponding sensor node. In this case, this maintenance was in the form of replacing the sensor board by a new empty one, which results in the sensor node reconfiguration process, via *OTAP*. When the *OTAP* process is complete, the S&A NETDEV changes its process state to ”Standby”, waiting for new task requests from other NETDEVs. The described behavior of the S&A NETDEV is represented on Figure 14.

B. Demonstration towards enhancing device with re-use and predictive maintenance capabilities

This demonstration consisted of validating scenarios regarding a Device NETDEV optimization models, analysis models for predictive maintenance and the re-use of components based on condition monitoring. Physically, it was used a welding machine, developed by TECHNAX, based on resistance welding. This machine performs welding jobs on forks and electrical components on a brush-holder (used in alternators). First, a metal sheet is punched out and bent into a work piece. Then, two different forks are positioned on one work piece. Finally, both forks and work piece are assembled. Sensors were used in predictive maintenance, namely 1) detection of welding process disturbances based on the electrodes temperature values and 2) automatic calibration of exposure time on an optical metrology system, promoted by INOS Hellas. Figure 15 represents the TECHNAX welding machine used in this demonstrator.

1) Predictive Maintenance: This scenario intends to validate how monitoring the electrode's temperature variation on a welding cycle and finding patterns between welding cycles can provide additional information regarding the electrode's wear out. Monitoring welding parameters give additional information in order to predict bad components exchange and also to decrease the machine down time. Figure 16 illustrates the electrodes' temperature variation on different welding cycles.

In this scenario, each welding cycle lasted around 200ms and the temperature sensors had a measurement cycle of approximately 4ms (250Hz). The sensor starts collecting temperature measurements only when the electrode temperature surpasses the room temperature. The sensor saves this reading as initial temperature of the process and, for the next 200ms, it collects every temperature reading. In the end of the welding cycle, the sensor collected around 50 measurements, which are sent to the gateway node. Using the I-RAMP[3] technology, the welding machine corresponds to a Device NETDEV and the sensor node, which contains a thermocouple, corresponds to a S&A NETDEV. The S&A NETDEV stores the received temperature readings. The Device NETDEV requests, via TDD, a *Measurement* task from the S&A NETDEV, which sends, via QRD, the sensor readings. The Device NETDEV processes these results and restarts the welding process.

2) Automatic Calibration: This scenario intends to validate how sensors can improve a metrology process. In this case, the metrology system is composed by cameras and lasers. The system is used in the welding process, by detecting patterns on a captured image, analysing if the forks are well positioned regarding the work piece. There are several ways external sensors can be applied to assist this type of metrology systems. One of them takes into account the luminosity conditions in which the measurement is being made. It is evident that the light conditions in the region where the measurement is being made can negatively influence the results, leading to either false positives or false negatives. Therefore, luminosity sensors are used to provide a reliable and effective feedback regarding the variation of light conditions. Figure 17 represents a screenshot of the analysis of two image captures of a fork

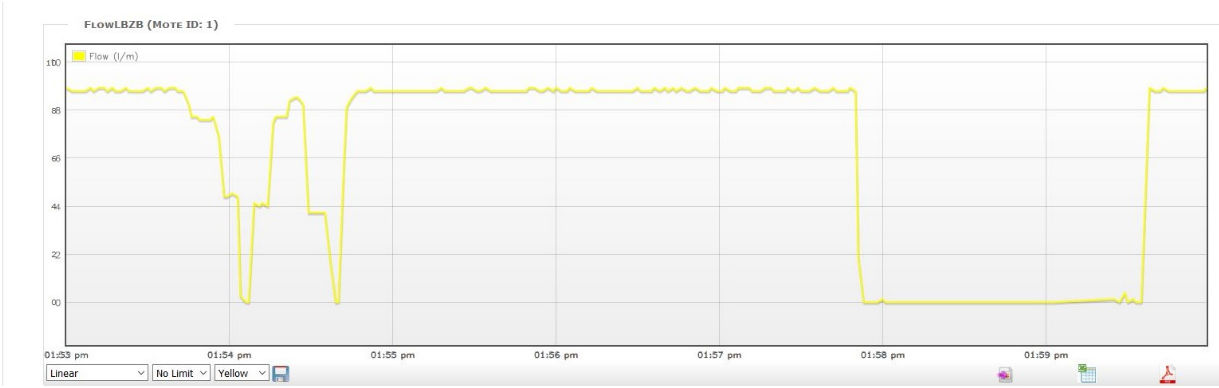

Figure 12. Water flow variation on PlugThings Framework.

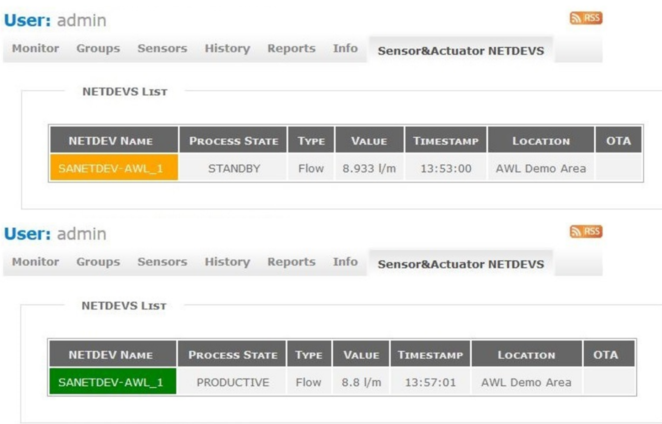

Figure 13. S&A NETDEV information on PlugThings Framework, before and after a *Measurement* task execution.

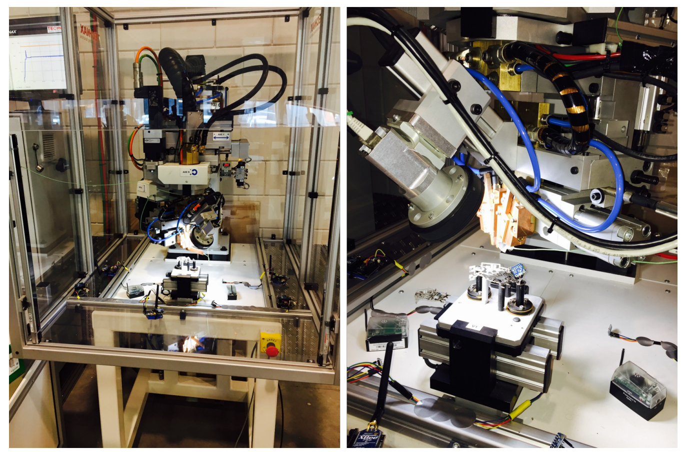

Figure 15. I-RAMP[3] demonstrator regarding enhancing device with re-use and predictive maintenance capabilities.

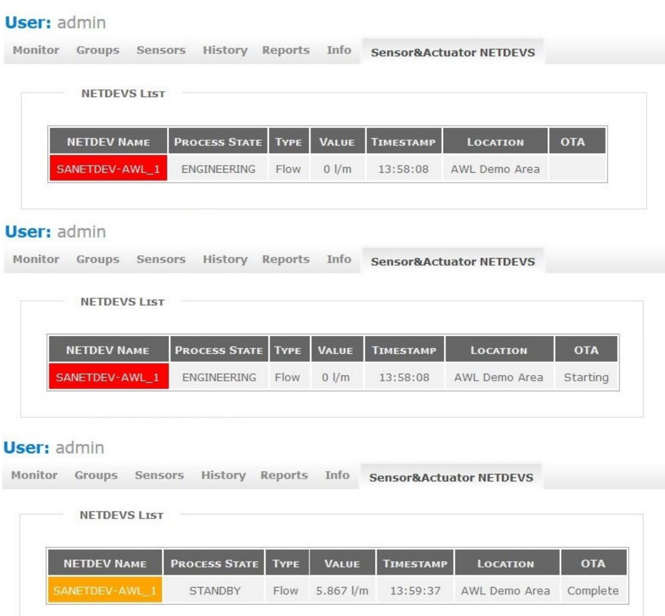

Figure 14. S&A NETDEV reconfiguration process when a sensor node malfunction is detected.

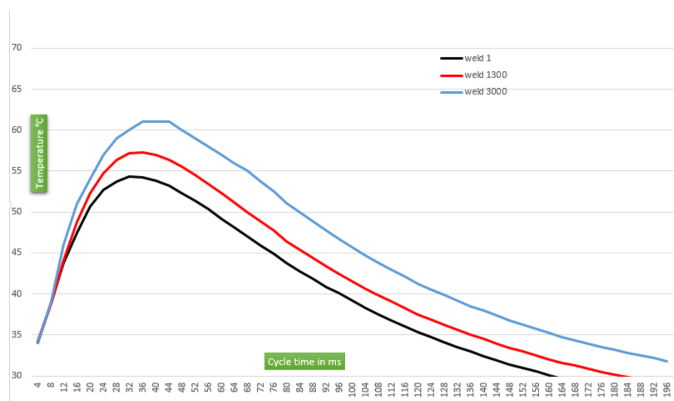

Figure 16. Electrodes' temperature variation on different welding cycles.

at the beginning of a welding process. The image on the left was taken without any exposure calibration, so detecting the fork is very difficult. On the other hand, the image on the right was taken with exposure calibration, so the fork was detected sucessfully.

The scenario is composed physically by a WSN, where each node is equipped with one luminosity sensor, a metrology system, composed by a camera and a laser and, the welding

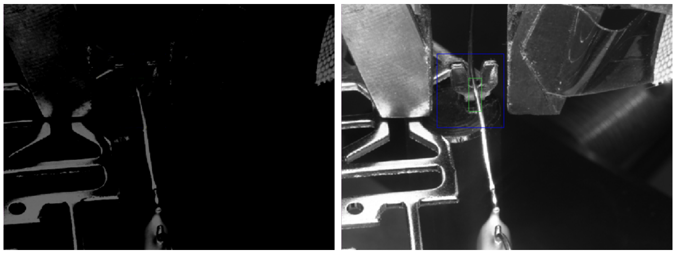

Figure 17. Metrology analysis of a non-calibrated and a calibrated camera image captures of a fork.

machine. The sensor nodes are distributed uniformly around the area to be monitored (the electrodes region). Using the I-RAMP[3] technology, each sensor node corresponds to one S&A NETDEV, the welding machine and the metrology system correspond both to Device NETDEVs. For easiness on explaining, they will be called TNX NETDEV for the welding machine and INOS NETDEV for the metrology system.

In order to start a new welding cycle, the TNX NETDEV requests the INOS NETDEV for fork presence, which starts by capturing an image, analyse it and send the result back. The success of the image analysis depends greatly on the quality of the image. To maintain a high quality on the image, the exposure time of the camera must be adjusted before taking the picture, according to the luminosity conditions (The darker the luminosity conditions are, the higher should be the exposure time). Therefore, before capturing the image, the INOS NETDEV must request a *Group Formation* task to one of the available S&A NETDEVs on the network. The S&A NETDEV should have capabilities for luminosity measurement and the corresponding sensor node should be physically located on the same machine as the metrology system is. The S&A NETDEV form a "luminosity" group, which is constantly monitored by a Process Analyzer NETDEV. If one of these sensors is not working properly or is down, the Process Analyzer can detect it by means of the Spatial Correlation technique.

VI. Discussion

As discussed several times throughout the present paper, the use of WSNs is referred as a key element on the I-RAMP[3] project. It has been explored as a benefit for the today's Manufacturing Systems by pushing forward the plug and produce concept and taking advantages of the latest technologies to do so. This plug and produce concept is achieved using the NETDEV concept on shop-floor equipment, which can readily describe and detail their own capabilities and announce themselves into the network to other NETDEV entities. The NETDEV technology allows collaboration between industrial equipments and execution of shop-floor tasks on demand. Therefore, the NETDEV technology delivers an easy and flexible solution for the industrial domain. Taking into account WSNs, all this flexibility and readiness is achieved making use of all the functionalities presented in previous sections, both at software and hardware level.

As described earlier, the collaboration between sensors, by means of *Group Formation* tasks is a way of reducing the communication entropy. This is important when several measurements from neighbor sensors need to be collected. Additionally, sensor collaboration aims for providing qualitative information (based on trend analysis) instead of quantitative (raw data). This means that the task requester does not need to have implemented any statistical techniques and doesn't make any extra sensor data processing, since this is done on the sensor side. Also, grouping sensors of different types and with different capabilities offers new sensing capabilities that are not available just with single sensors. Sensor collaboration is allied with the sensor location functionality, which allows to know, with a certain degree of precision, the location of sensors in a restricted area. Automatic sensor location influences and guides how sensors should organize and collaborate among themselves, ensuring the system reliability and effectiveness.

Sensor diagnosis capabilities provides feedback about the health conditions of the WSN, making use of sensor validation techniques already explored in the literature and tested in manufacturing environments. This diagnosis ensures a better monitoring of the manufacturing process and, therefore, is extremely beneficial in the prediction of maintenance phases. Also, the mechanism to detect malfunction sensors inside a sensor group is able to exclude a malfunction sensor node and replacing it by a new available one, without pausing the task execution. Maintenance can be done prior to the task execution. If redundancy on the number of sensors is not possible, sensor reconfiguration capabilities present a way to quickly react and resolve the diagnosed sensor malfunction, which may become a bottleneck on the production line. Recovery from failures offers more flexibility to the maintenance process, without requiring high-qualified technical personnel.

Another advantage of this approach is related with all the functionalities already available from a dedicated framework, releasing the user from being concerned about sensor collaboration and data processing. The only concern is the sensor integration, which is done using the S&A NETDEV template solution. From that point, information can be easily accessed, monitored and diagnosed. Thus, it is not required for the final user to know in detail how to implement a WSN diagnostics system. Instead, the focus should be on what to do when a certain malfunction has occurred and how to relate sensor group information with the product life-cycle in terms of process parameterization. This advantage is enhanced with the automatic process of forming, deforming and reacting to sudden changes in a sensor group, based on a certain task parameters and sensor location. Since the communication between NETDEV entities is based on a standardized task-driven XML language – DIL - it is very easy to implement a new system that encapsulates a machine, capable of communicating with these entities and easily interpret sensor information for process monitoring.

Additionally, this approach presents a self-reconfiguration capability when facing sudden sensor breakdown. In a real manufacturing environment situation, the shop-floor operator only needs to find the malfunctioning sensor node (information already provided by the Process Analyzer NETDEV) and replace its hardware by a new one. Automatically, the sensor node reconfigures itself by being re programed over the air, sparing the user to remove the sensor node, connects the board to a PC and writes or rewrites any lines of code.

In the perspective of the Manufacturing System Designer, there are benefits regarding the application of WSNs in industry. Based on the fact that most manufacturing environments are currently using wired sensors solutions instead of WSNs, the cabling complexity and savings in terms of installation time and cost can be reached, avoiding connecting and configuring sensors on PLCs or Machine Controllers. Wireless is a preferable solution, because of the amount of sensors used and their (sometimes) harsh locations on the shop-floor, and the effort associated with changing the code in the PLC to configure a new sensor. The easiness to integrate a new sensor into the system is achieved by only switching on a sensor node, using the plug and produce concept, which is automatically recognized as a S&A NETDEV, becoming ready for use. The plug and produce concept allows a rapid reaction to any foreseen and unforeseen events, such as sensor replacement, sensor addition for redundancy purposes in critical environments or in the case of sensor removal when disassembling a production line.

However, despite all the benefits that the plug and produce concept bring to the shop-floor, and based on the fact that most of the manufacturing systems nowadays have stable and functional production methods, there is mandatory to make, first, a risk assessment to evaluate how security issues associated with the new technology can threaten the company by making it venerable and, second, a coherent business plan to assess the impact of such a technology installation in the production line.

Security is a big issue in the industrial sector, specially nowadays, where networked infrastructures are used to connect several manufacturing systems. Companies are very concern to protect both their private data and know-how. Actually, methods for security can be in the form of entity identification and authentication for multi-level access control, network and communication protection, privacy, trust and management of information, and system fault tolerance. Regarding the use of industrial equipment, specially connected equipment such as WSNs, extended with the I-RAMP[3] technology, security concerns can be identified at the physical level, which includes equipment and communication, and cyber level, which includes the NETDEV infrastructure.

Zia and Jain [62], [63] pointed out that sensor networks pose unique security challenges because of their computing and communication limitations. Since security approaches require a certain amount of resources, such as data memory, storage space and energy to power the sensor node, it is difficult to implement such approaches in a WSN. It would be highly expose to remote tempering of it's sensor nodes, since the communication protocol may not be robust enough to guaranty a reliable communication. Moreover, WSNs may be deployed in environments where their sensor nodes would be prone to physical tempering, such as capture and vandalism. Sert [64] identified and described various attack types that are frequent in WSNs and corresponding defense mechanisms.

Regarding the NETDEV technology, security is directly associated with the network security level and the infrastructure that supports the NETDEVs, in this case, the UPnP protocol. Wong [65] identified many security threats on a MAS, such as corrupted naming and matchmaking services, insecure communication channel, insecure delegation and lack of accountability. Although many approaches [65], [66] tried to implement a security infrastructure for MAS, security itself should be a design characteristic in protocols and frameworks used for agent-oriented systems. In the case of the UPnP, this protocol was build on the assumption that the network is secure and only trusted devices have access to it. Selén [67] points out that the UPnP architecture currently provides one solution for security [68]. This solution proposes a relatively complex set of protocols and procedures, which make designing a secure UPnP-compliant device both expensive and time-consuming. Also, the plug and play capabilities of the protocol become jeopardized, since this scheme requires active human interaction, where the user must enter a password for entity authentication. More recently, Pehkonen [69] proposed and applied a patent for a secure UPnP network architecture.

A more business oriented perspective was taken in [70], where some aspects such as the impact of technology installation, human resources and training were discussed. The authors refer that such a technology would require an initial investment in human resources for machine adaptation and to fine tune the NETDEVs concept to be compliant with the overall manufacturing system, along with the required time for testing and validation. Additionally, training the system's end-users would also be required, along with a reformulation of the shop-floor action protocols. Nevertheless, the authors point out that not every equipment on a production line needs to be adapted. Possibly, only the more critical machines for maintenance or with high number of ramp-up phases should be shielded with the NETDEV, taking advantage of all inherent functionalities of the technology. This way, a previous analysis should be made, regarding the relation of the amount of machines that can be virtualized as a NETDEV against the long-term required investment. In that sense, this kind of system adaptation is more preferable to be applicable in machines with an expected long-term use, like several years, rather than a couple of months, due to the difficulty to overcome the CAPEX required to achieve the companies goals of performance and effectiveness. Although all these advantages could be achieved without the NETDEV concept, they would require a significantly higher effort for installation and maintenance. Therefore, to achieve the same level of efficiency and effectiveness, it would be necessary a considerably higher CAPEX and OPEX throughout a long time-span. For this reason, the NETDEV technology is more appealing and definitely a better cost-benefit approach.

VII. Roadmap for the Future

Flexibility is seen as one of the key aspects for bringing manufacturing back to Europe on a large scale basis. The tendency to more customized products and demand-driven manufacturing processes is seen as an opportunity for Europe's manufacturing industry – and in particular the Small and Medium-sized Enterprises (SMEs) – to respond to the nowadays leading roles, mainly of Asian companies in mass-oriented production. In order to meet the challenges that come up with rapid changing product portfolios, smaller lot sizes and continuously evolving process technologies, manufacturing systems are required to be easily upgradeable, into which new technologies and new functions can be readily integrated [71]. Demands for increasing productivity through highly optimized production processes create the need for novel manufacturing control systems, which are required to manage product and production variability and disturbances effectively and efficiently [17], and to implement agility, flexibility and reactivity.

In order to meet these challenges, high efforts have been made and are still to be done in research for flexible and agile manufacturing systems. Significant improvements in reconfiguration, performance or disturbance handling have been shown. However, the large-scale adoption in the industry is still missing. Among others, the major barriers are seen in proprietary tools, equipment and software systems as well as missing standards of usually heterogeneous manufacturing environments [16]. A coexistence of classical and advanced technologies as well as the stepwise approach is required in order to introduce new technology successfully [72].

Even more important for a broad take-up of new technology in industry is the technological feasibility and industrial readiness as well as a consequent cooperation of end users, system and components suppliers and research, in order to minimize the risks on investments and to gain acceptance at the end users side [73].

As a consequence, an integrated approach is required in order to tackle the challenges of wide technology uptake. Industrial leadership and the reflection of the complete added-value chain is needed. The requirements and opinions of big end users are very important as they finally use such flexible systems in their factories and thus, need to be taken into account in a very early stage. However, a very specific technological development for one or a very limited number of end users as a part of the I-RAMP^3 consortium is not efficient in order to guarantee a broad technology uptake. This would probably limit the technology development and deployment to the specific needs of the respective end users and hamper a wider penetration on the market. Much more promising is an approach in which the component and equipment suppliers as well as the system integrators play a leading role. This group is predesignated to overcome the limitations as they provide a huge technological basis for a huge number of application areas. They are usually in close contact to the end users and are often involved in the end users' strategic planning for innovation management [74]. SMEs are in particular important as they are the key driver of innovation in Europe [75]. Enabling the SMEs for the development and production of flexible systems for a broad number of applications, would therefore form the basis of a wide technology uptake in the industry.

Of essential importance is a widespread technology basis rather than only a punctual implementation and integration of technology. Several application areas and industrial sectors need to be address in parallel in order to achieve a critical mass of deployment. Standardization and harmonization needs to be taken into account and approaches for a smooth integration of legacy systems need to be developed. Several research studies demonstrated, based on concrete data and evidence, the benefits and role of standards in supporting and driving research and innovation. Standards play a multiple, catalytic role in the innovation system and in research projects by providing common terminologies, harmonised methodologies and comparability between research activities. At the same time, standards have a particular importance for market acceptance of technology-based innovation, by improving the marketability of research and innovation results.

VIII. CONCLUSIONS

Innovative intelligent systems have driven technology for years, and industry has followed this track. This enables manufacturing processes to improve their reliability and responsiveness when facing production changes or downtime, as well as time efficiency to minimize costs and effectiveness to increase production quality. These objectives guided the NETDEV development. Intelligent functionalities, such as information sharing by inter-device communication, device self-description, collaboration and negotiation, process optimization and condition monitoring are inherent capabilities of the NETDEVs. Applied to sensor & actuators, these capabilities are extended to self-organization, automatic device location, WSN health monitoring with diagnosis and automatic reconfiguration. Therefore, by taking advantage of these functionalities, one can greatly influence industrial processes regarding the decrease of ramp-up time, as well as the time needed to recover from scheduled and unscheduled maintenance. These results are a competitive advantage in demanding and fluctuating contexts. The main developments presented throughout the paper depict that, in terms of WSNs applicability in industry, there are open opportunities to explore new solutions and to improve the currently used systems. Despite the benefits of all functionalities presented in this paper, the boost of the deployment of smart components developed within I-RAMP^3 technology is needed, by unifying efforts towards the swift standardization of NETDEVs. The acceptance of NETDEVS and WSNs into industrial context needs to be pushed forward, by creating a developers' community and performing more pragmatic and real test-case demonstrators. The present work is a clear step forward into a reliable and flexible approach for industrial WSNs, aiming for paving the way into more intelligent manufacturing systems.

ACKNOWLEDGMENT

This research was supported by project I-RAMP^3 (FoF.NMP.2012-3) – Intelligent Network Devices for fast Ramp-up – funded by the European Commission under the Seventh Framework Programme for Research and Technological Development. We would like to show our gratitude to the whole I-RAMP3 consortium for all the support, assistance and fruitfull collaboration that resulted in the all the achievements and technological advances depicted in the present paper.

REFERENCES

[1] R. Pinto, J. Reis, R. Silva, V. Sousa, and G. Gonçalves, "Self-organising smart components in advanced manufacturing systems," in INTELLI 2015, The Fourth International Conference on Intelligent Systems and Applications. IARIA, 2015, pp. 157–163.

[2] R. Pinto, J. Reis, V. Sousa, R. Silva, and G. Gonçalves, "Self-diagnosis and automatic configuration of smart components in advanced manufacturing systems," in INTELLI 2015, The Fourth International Conference on Intelligent Systems and Applications. IARIA, 2015, pp. 164–169.

[3] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards," Computer communications, vol. 30, no. 7, 2007, pp. 1655–1695.

[4] H. Ramamurthy, B. Prabhu, R. Gadh, and A. M. Madni, "Wireless industrial monitoring and control using a smart sensor platform," Sensors Journal, IEEE, vol. 7, no. 5, 2007, pp. 611–618.

[5] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," Industrial Electronics, IEEE Transactions on, vol. 57, no. 12, 2010, pp. 4219–4230.

[6] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends," sensors, vol. 9, no. 6, 2009, pp. 4728–4750.

[7] J. J. Evans, "Wireless sensor networks in electrical manufacturing," in Electrical Insulation Conference and Electrical Manufacturing Expo, 2005. Proceedings. IEEE, 2005, pp. 460–465.

[8] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," Industrial Electronics, IEEE Transactions on, vol. 56, no. 10, 2009, pp. 4258–4265.

[9] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Distributed collaborative control using wireless sensor and actuator networks," in Future Generation Communication and Networking, 2008. FGCN'08. Second International Conference on, vol. 1. IEEE, 2008, pp. 3–6.

[10] P. Neumann, "Communication in industrial automation—what is going on?" Control Engineering Practice, vol. 15, no. 11, 2007, pp. 1332–1347.

[11] K. Xin, X. Cao, J. Chen, P. Cheng, and L. Xie, "Optimal controller location in wireless networked control systems," International Journal of Robust and Nonlinear Control, vol. 25, no. 2, 2015, pp. 301–319.

[12] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The adacor evolution," Computers in Industry, vol. 66, 2015, pp. 99–111.

[13] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," Journal of manufacturing systems, vol. 29, no. 4, 2010, pp. 130–141.

[14] J. Ferber, Multi-agent systems: an introduction to distributed artificial intelligence. Addison-Wesley Reading, 1999, vol. 1.

[15] S. Deen, "Agent-based manufacturing: Advances in the holonic approach, 2003."

[16] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," Engineering Applications of Artificial Intelligence, vol. 22, no. 7, 2009, pp. 979–991.

[17] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: Prosa," Computers in industry, vol. 37, no. 3, 1998, pp. 255–274.

[18] S. Liu, W. A. Gruver, D. Kotak, and S. Bardi, "Holonic manufacturing system for distributed control of automated guided vehicles," in Systems, Man, and Cybernetics, 2000 IEEE International Conference on, vol. 3. IEEE, 2000, pp. 1727–1732.

[19] L. Mönch, M. Stehli, and J. Zimmermann, "Fabmas: An agent-based system for production control of semiconductor manufacturing processes," in Holonic and Multi-Agent Systems for Manufacturing. Springer, 2003, pp. 258–267.

[20] M. Fletcher, D. McFarlane, A. Lucas, J. Brusey, and D. Jarvis, "The cambridge packing cell—a holonic enterprise demonstrator," in Multi-Agent Systems and Applications III. Springer, 2003, pp. 533–543.

[21] P. Leitão and F. Restivo, "Adacor: A holonic architecture for agile and adaptive manufacturing control," Computers in industry, vol. 57, no. 2, 2006, pp. 121–130.

[22] D. McFarlane et al., "Industrial adoption of agent-based technologies," IEEE Intelligent Systems, no. 1, 2005, pp. 27–35.

[23] S. Bussmann and K. Schild, "Self-organizing manufacturing control: An industrial application of agent technology," in MultiAgent Systems, 2000. Proceedings. Fourth International Conference on. IEEE, 2000, pp. 87–94.

[24] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," Industrial Electronics, IEEE Transactions on, vol. 53, no. 1, 2006, pp. 322–337.

[25] F. P. Maturana, R. Staron, K. Hall, P. Tichỳ, P. Šlechta, and V. Mařík, "An intelligent agent validation architecture for distributed manufacturing organizations," in Emerging Solutions for Future Manufacturing Systems. Springer, 2005, pp. 81–90.

[26] W. Shen, Q. Hao, H. J. Yoon, and D. H. Norrie, "Applications of agent-based systems in intelligent manufacturing: An updated review," Advanced engineering INFORMATICS, vol. 20, no. 4, 2006, pp. 415–431.

[27] T. Ribeiro and G. Gonçalves, "Formal methods for reconfigurable assembly systems," in Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on. IEEE, 2010, pp. 1–6.

[28] GNOSIS Consortium, "Knowledge Systematization: Configuration Systems for Design and Manufacturing," p. 26, 2000. [Online]. Available: http://www.ims.org/wp-content/uploads/2011/11/2.4.12.2-Final-Report-GNOSIS.pdf

[29] J. Peschke, A. Lüder, and H. Kühnle, "The pabadis'promise architecture-a new approach for flexible manufacturing systems," in Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on, vol. 1. IEEE, 2005, pp. 6–pp.

[30] G. Gonçalves, J. Reis, R. Pinto, M. Alves, and J. Correia, "A step forward on intelligent factories: A smart sensor-oriented approach," in Emerging Technology and Factory Automation (ETFA), 2014 IEEE. IEEE, 2014, pp. 1–8.

[31] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," ACM Transactions on Sensor Networks (TOSN), vol. 6, no. 3, 2010, p. 23.

[32] J. Ravichandran and A. I. Arulappan, "Data validation algorithm for wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 2013, 2013.

[33] Y. Shang and W. Rum, "Improved mds-based localization," in INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4. IEEE, 2004, pp. 2640–2651.

[34] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in Proceedings of the 7th annual international conference on Mobile computing and networking. ACM, 2001, pp. 166–179.

[35] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, vol. 5. IEEE, 2001, pp. 2926–2931.

[36] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in Proceedings of the 7th annual international conference on Mobile computing and networking. ACM, 2001, pp. 1–14.

[37] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the radar user location and tracking system," technical report, Microsoft Research, Tech. Rep., 2000.

[38] K. Vandenbussche et al., "Fine-grained indoor localisation using wireless sensor networks," Signal, vol. 75, 2005, p. 70.

[39] T. Singal, Wireless communications. Tata McGraw-Hill Education, 2010.

[40] C. Levis, J. T. Johnson, and F. L. Teixeira, Radiowave propagation: physics and applications. John Wiley & Sons, 2010.

[41] A. F. Molisch, Wireless communications. John Wiley & Sons, 2007.

[42] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay et al., "A macroscope in the redwoods," in Proceedings of the 3rd international conference on Embedded networked sensor systems. ACM, 2005, pp. 51–63.

[43] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Out-of-the-box environmental monitoring," in Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on. IEEE, 2008, pp. 332–343.

[44] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, T. Harmon, C. Harvey, J. Jay, E. Kohler, S. Rothenberg, and M. Srivastava, "Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks," Center for Embedded Network Sensing, 2006.

[45] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 2004, pp. 214–226.

[46] N. Branisavljević, Z. Kapelan, and D. Prodanović, "Improved real-time data anomaly detection using context classification," Journal of Hydroinformatics, vol. 13, no. 3, 2011, pp. 307–323.

[47] J.-L. Bertrand-Krajewski, J.-P. Bardin, M. Mourad, and Y. Beranger, "Accounting for sensor calibration, concentration heterogeneity, measurement and sampling uncertainties in monitoring urban drainage systems," Water Science & Technology.

[48] O. J. Mengshoel, M. Chavira, K. Cascio, S. Poll, A. Darwiche, and S. Uckun, "Probabilistic model-based diagnosis: An electrical power system case study," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 40, no. 5, 2010, pp. 874–885.

[49] O. J. Mengshoel, A. Darwiche, and S. Uckun, "Sensor validation using bayesian networks," in Proc. 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS-08), 2008.

[50] J. Chen, H. Li, D. Sheng, and W. Li, "A hybrid data-driven modeling method on sensor condition monitoring and fault diagnosis for power plants," International Journal of Electrical Power & Energy Systems, vol. 71, 2015, pp. 274–284.

[51] A. Messai, A. Mellit, I. Abdellani, and A. M. Pavan, "On-line fault detection of a fuel rod temperature measurement sensor in a nuclear reactor core using anns," Progress in Nuclear Energy, vol. 79, 2015, pp. 8–21.

[52] S. Dauwe, D. Oldoni, B. De Baets, T. Van Renterghem, D. Botteldooren, and B. Dhoedt, "Multi-criteria anomaly detection in urban noise sensor networks," Environmental Science: Processes & Impacts, vol. 16, no. 10, 2014, pp. 2249–2258.

[53] K. Zhang, S. Shi, H. Gao, and J. Li, "Unsupervised outlier detection in sensor networks using aggregation tree," in Advanced data mining and applications. Springer, 2007, pp. 158–169.

[54] J. Rossiter and Z. Hensley, "Sensor data management, validation, correction, and provenance for building technologies," ASHRAE Transactions, vol. 120, 2014, p. 370.

[55] B. Iglewicz and D. C. Hoaglin, How to detect and handle outliers. Asq Press, 1993, vol. 16.

[56] G. Olsson, M. Nielsen, Z. Yuan, A. Lynggaard-Jensen, and J.-P. Steyer, "Instrumentation, control and automation in wastewater systems," Water Intelligence Online, vol. 4, 2005, p. 9781780402680.

[57] M. Mourad and J. Bertrand-Krajewski, "A method for automatic validation of long time series of data in urban hydrology," Water Science & Technology, vol. 45, no. 4-5, 2002, pp. 263–270.

[58] L. Jayashree, S. Arumugam, and A. Meenakshi, "A communication-efficient framework for outlier-free data reporting in data-gathering sensor networks," International Journal of Network Management, vol. 18, no. 5, 2008, pp. 437–445.

[59] I. Adly, H. Ragai, A. El-Hennawy, and K. Shehata, "Over-the-air programming of psoc sensor interface in wireless sensor networks," in MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference. IEEE, 2010, pp. 997–1002.

[60] Libelium, "Waspmote Technical Guide," p. 168, 2013. [Online]. Available: http://www.libelium.com/uploads/2013/02/waspmote-technical_guide_eng.pdf

[61] ——, "Over the Air Programming with 802.15.4 and ZigBee: Laying the groundwork," p. 40, 2013. [Online]. Available: http://www.libelium.com/uploads/2013/02/over_the_air_programming.pdf

[62] T. Zia and A. Zomaya, "Security issues in wireless sensor networks," in Systems and Networks Communications, 2006. ICSNC'06. International Conference on. IEEE, 2006, pp. 40–40.

[63] M. K. Jain, "Wireless sensor networks: Security issues and challenges," International Journal of Computer and Information Technology, vol. 2, no. 1, 2011, pp. 62–67.

[64] S. A. Sert, E. Onur, and A. Yazici, "Security attacks and countermeasures in surveillance wireless sensor networks," in Application of Information and Communication Technologies (AICT), 2015 9th International Conference on. IEEE, 2015, pp. 201–205.

[65] H. C. Wong and K. Sycara, "Adding security and trust to multiagent systems," Applied Artificial Intelligence, vol. 14, no. 9, 2000, pp. 927–941.

[66] G. Beydoun, G. Low, H. Mouratidis, and B. Henderson-Sellers, "A security-aware metamodel for multi-agent systems (mas)," Information and Software Technology, vol. 51, no. 5, 2009, pp. 832–845.

[67] K. Selén, "Upnp security in internet gateway devices," in TKK T-110.5190 Seminar on Internetworking, 2006.

[68] C. Ellison, "Upnp security ceremonies version 1.0," in UPnP Forum, 2003.

[69] V. Pehkonen and J. Koivisto, "Secure universal plug and play network," in Information Assurance and Security (IAS), 2010 Sixth International Conference on. IEEE, 2010, pp. 11–14.

[70] M. Kasperczyk and E. Ridders, "Efficient implementation of network-enabled devices into industrial environment," in INTELLI 2015, The Fourth International Conference on Intelligent Systems and Applications. IARIA, 2015, pp. 148–149.

[71] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: key to future manufacturing," Journal of Intelligent manufacturing, vol. 11, no. 4, 2000, pp. 403–419.

[72] C. Gerber, H.-M. Hanisch, and S. Ebbinghaus, "From iec 61131 to iec 61499 for distributed systems: a case study," EURASIP Journal on Embedded Systems, vol. 2008, 2008, p. 4.

[73] K. Sundermeyer and S. Bussmann, "Introduction of agent technology into a manufacturing company-experiences from on industrial project," Wirtschaftsinformatik, vol. 43, no. 2, 2001, pp. 135–+.

[74] Siemens PLM Software, "Leveraging suppliers for strategic innovation," 2009. [Online]. Available: http://www.plm.automation.siemens.com/

[75] EuroStat, "Industry and services statistics introduced," 2015. [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Industry and_services_statistics_introduced