Responding to Failure Events in the Manufacturing Environment

Christos Anagnostopoulos^{1,}, Christos Alexakos¹, Apostolos Fournaris¹, Christos Koulamas¹, and Athanasios Kalogeras¹*

¹Industrial Systems Institute, 26504 Platani Patras, Greece

Abstract. The manufacturing environment is characterized by increased complexity with different devices, systems and applications that need to interoperate, while residing at different layers of the classical industrial environment hierarchy. The introduction of the Industrial Internet of Things with increasingly smarter devices drives towards flatter hierarchies. This paper deals with an architecture for integration of IIoT devices in the manufacturing environment utilizing a Multi Agent System to this end. This extended architecture is utilised so as to perform failure detection of both IIoT devices and manufacturing resources, and react by altering the manufacturing process either automatically or semi-automatically.

1 Introduction

The manufacturing environment is characterized by increased complexity with different types of devices, systems and applications that need to interoperate, while residing at different layers of the classical industrial environment hierarchy. With reference to the classical hierarchy scheme, the industrial environment systems and applications may reside on three layers: the Shop Floor layer being closer to the production process and dealing with controller devices (PLCs, NCs, robot controllers) as well as sensing and actuating devices, the Manufacturing Execution System layer being a middle layer dealing among others with manufacturing resource management and life cycle management, and the Enterprise Resource Planning (ERP) layer being a top layer dealing with order placement and providing interfaces to enterprise systems.

Having increasingly smarter devices and devices with increased computational resources integrated in the manufacturing environment drives a trend for moving from rigid hierarchical schemes towards a flatter hierarchy and increasing distribution of the manufacturing systems and processes. In this context the two lower layers of the classical manufacturing hierarchy, the Shop Floor layer and the Manufacturing Execution System (MES) layer, and their processes may be completely distributed and decentralized. Such an approach has been followed by the PABADIS’PROMISE (P2) project that substituted the centralised MES layer by a Multi Agent System (MAS) architecture [1]. This architecture envisaged different types of agents with different roles that allowed the efficient distribution of MES functionalities among them.

* Corresponding author: anagnostopoulos@isi.gr

The Industrial Internet of Things (IIoT) is expected to revolutionize manufacturing while influencing different domains pertaining to almost all sectors of human life and activity [2]. It represents the application of the Internet of Things (IoT) in the manufacturing domain, leading to a fourth industrial revolution (Industry 4.0) and being associated with huge amounts of data aggregated. It can reap benefits for the industry through greater operational efficiency, improved safety and preventive maintenance of assets.

The present paper is based on previous work extending the P2 architecture towards integrating IIoT devices [3]. Our proposed system relies on and utilizes this extended architecture in order to perform failure detection of manufacturing resources or IIoT devices and react by automatically or semi-automatically altering the manufacturing process. The failure event detection method followed [4] is extended and generalized.

The paper structure is as follows. Section 2 presents prior work that this paper is based on. Section 3 presents the presented system architecture detailing its components. Finally, Section 4 presents conclusions and discussion.

2 Prior Work

Cyber Physical Systems make possible the synchronization and monitoring of information between physical systems and the cyber computational space [5]. In this context the manufacturing industry is driven to a new revolution, namely Industry 4.0 [6]. A characteristic of this evolution is decentralization, i.e. moving towards flatter hierarchies with reference to the rigid hierarchical schemes of the past in the manufacturing environment.

The PABADIS’PROMISE (P2) project contributed towards this end at a relatively early stage. P2 architecture extended across all layers of the classical hierarchy scheme prevalent at the manufacturing environment, from the Enterprise Resource Planning (ERP) layer on the top, through the Manufacturing Execution Systems (MES) layer in the middle, down to the Shop Floor Layer. A complete decentralization of the MES layer was envisaged increasing autonomy and flexibility by employing a Multi Agent System (MAS) following the concepts of the GAIA methodology [7]. Semantic data interoperability is addressed by the P2Ontology [8] dealing with the necessary semantics involved in the P2 MAS paradigm. Data exchange is done by the Product and Production Process Description Language (P5DL) [9] different representation languages.

P2 architecture was extended towards efficient integration of IIoT devices by the VISETAK system [3]. The P2 MAS was extended by adding a further Agent Role, the role of Sensor Agent, so as to deal with effective IIoT device integration. This integration is made possible by assigning each group of IIoT devices a Sensor Agent, being responsible for data exchange between the P2 MAS and the IIoT devices utilizing an intermediate middleware to this end. The VISETAK MAS is depicted in Figure 1.

The idea of the P2 MAS is maintained in VISETAK MAS, being to map the requested capabilities by the different orders placed at the ERP layer with the available abilities of the resources of the Shop Floor layer. Resource Agents expose these abilities to the MAS middleware. When an order is placed at the ERP level, it is assigned to the Order Agent Supervisor. Product Data Repository information is retrieved and the necessary Order Agents are created. The Ability Broker is responsible for performing the mapping between Order Agents and Resource Agents allocating the necessary resources to each order that is placed in the system.

The Sensor Agents enable the integration of IIoT device data to the overall system. The semantics of this data is dealt with through the adoption of the Semantic Sensor Network Ontology (SSN) [10].

Fig. 1. VISETAK MAS.

Aforementioned Industry 4.0 and the Industrial Internet Consortium (IIC) [11] represent the two main platforms that deal with the IIoT integration in the manufacturing domain. Both present Reference Architectures in order to make possible the IIoT adoption. The Reference Architecture Model for Industry 4.0 (RAMI 4.0) [12] and the Industrial Internet Reference Architecture (IIRA) [13] present the two relevant efforts, while there is recent effort for their mapping and alignment [14]. Our presented work maps on RAMI 4.0 architectural axis as well as IIRA functional viewpoint.

Different technologies are associated with the IIoT introduction. Multicast DNS (mDNS) [15] offers resource and service discovery in small networks missing a conventional DNS server. Constrained Application Protocol (CoAP) [16] offers a Machine-to-Machine application web transfer protocol designed for constrained nodes and constrained networks. Message Queue Telemetry Transport (MQTT) [17] is a Machine-to-Machine extremely lightweight publish / subscribe messaging transport protocol. Finally, Advanced Message Queuing Protocol (AMQP) [18] also employs asynchronous publish / subscribe over TCP.

3 System Architecture

The proposed system architecture is presented in Figure 2. The architecture builds on top and improves the VISETAK system architecture depicted in Figure 1. The former architecture utilized http gateways for the needs of integration of IIoT devices into the VISETAK MAS. This approach certainly induced overhead in the overall data transmission as transmission times between gateways was increased. A further drawback was associated with the increase in the Sensor Agent computational time as the same IIoT device data might be needed by different systems and applications.

The presented system architecture takes care of these drawbacks introducing the IIoT collector, being a lightweight software implementation acting as a middleware between the IIoT devices and the MAS. This approach reduces significantly overall overhead. The presented system architecture complies with the current state-of-the-art moving intelligence to the “edge” of the network [19]. Thus, it helps remove heavy protocol stacks in the interaction of sensing devices with the MAS through the Message Queue (MQ) Server.

Fig. 2. System Architecture.

3.1 IIoT Devices

A range of embedded devices with different capabilities has been utilized at the “edge” of the system. Wireless connectivity has been used and RESTful Web Services over HTTP with JSON content format. URIs are used for sensor identification, allowing multi-level access both directly through IPv6 and via embedded gateways translating HTTP/JSON to CoAP/CBOR equivalent.

All IIoT sensors execute sampling, storage, feature extraction, classification and novelty detection services. HTTP or CoAP accessed set of resource URIs makes possible their external control. Their most important resources are the following:

- /info: set of not configurable parameters that depends on the actual implementation, including buffer size, maximum number of buffers, etc
- /config: set of embedded detection engine configurable parameters, including sampling rate, sliding window for acquired signal processing size and step, etc
- /stateset: set of configurable state descriptions and thresholds, in order to generate a novelty event
- /event: resource of the last event triggered by the node.

It is possible to classify sensor nodes to three classes:

- Simple scalar sensors can be sustained by low bandwidth networking technologies and resource constrained embedded devices, as their processing and sampling bandwidth requirements are not high
- High quality, multiple axis vibration sensors present an extreme case with reference to power and processing resources, dealing with complex data streams
- A third class comprises nodes that involve series and vector measurements but with lower sampling rates and reduced accuracy needs. Mid processing power hardware with adequate memory can support this case.

A solution with off the shelf components has been followed to implement nodes of the third class. The relevant structure is shown in Figure 3.

Fig. 3. Class 3 Embedded node structure.

A Freescale FRDMK64F embedded board has been used for sampling, signal processing, and detection. A CC2538 based Openmote board has been used for the implementation of IEEE-802.15.4 wireless interface as well as IPv6, RPL and CoAP. An Xbee compatible shield has been used for the two board communication.

With reference to the second class an embedded PC or tablet device has been used to control the sensor node based on a DT9837B USB system. HTTP/JSON Web-Services based interfaces to IIoT have been developed. The relevant structure is depicted in Figure 4.

Fig. 4. Class 2 Embedded node structure.

3.2 IIoT Data

IIoT integration requires that the events generated by IIoT devices are transmitted to the manufacturing environment in due time to take the necessary action. IIoT devices have their own networking infrastructure, thus there is a need for data aggregation. This task is dealt with by the iIoT-Collect and the MQ Server.

iIoT-Collect is a python and MySQL implemented lightweight platform supporting HTTP and CoAP. It comprises three parts: a GUI for user administration services, a scheduler module that is responsible for checking for devices with retrieval action in the next time frame, and an invoker module that invokes relevant device services and submits responses to the MQ Server.

The MQ Server is implemented using open source Rabbit MQ message queue server. It maintains one queue per IIoT device per consuming system. In the case that we deal with there is a need for one queue consumed by the MAS Sensor Agent, which is the point where the MAS is informed about any events associated with the IIoT device, and for a second queue consumed by a script saving data to a NoSQL database.

JSON formatted messages arriving in a MQ Server queue subscribed by a Sensor Agent are converted into JAVA classes and parsed to P2Ontology and SSN Ontology. Thus they

become MAS understandable. Then the Sensor Agent processes the information sent by the IIoT device. If this information is related to some failure leading to missing some ability in the Shop Floor, the Resource Agent Supervisor is notified. This is then propagated to the relevant Resource Agent that notifies related Order Agents for lack of the ability. Then, a negotiation takes place so that the Order Agents find alternative resources for the missing ability.

4 Conclusions

This paper presents a system architecture for IIoT integration in the manufacturing environment. The approach builds on top of previous work enhancing a MAS based architecture with a middleware designed for efficiently integrating IIoT devices utilising moving intelligence to the “edge” of the network, near the IIoT devices. This lightweight middleware reduces overhead with reference to the integration with the MAS system. Upon the occurrence of an event the MAS responds by possibly re-orchestrating related manufacturing process workflows. Testing and validation of the presented approach is being done by means of a manufacturing environment simulator.

Acknowledgement

We acknowledge support of this work by the project “I3T - Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments” (MIS 5002434) which is implemented under the “Action for the Strategic Development on the Research and Technological Sector”, funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

1. A.P. Kalogeras, L. Ferrarini, A. Lueder, J. Gialelis, C. Alexakos, J. Peschke, and C. Veber, *Ontology-driven Control Application Design Methodology*, *Proceedings of the 12th IEEE Conference on Emerging Technologies and Factory Automation – ETFA '2007*, Patras, Greece, September 25-28 (2007)
2. D. Serpanos, M. Wolf *Industrial Internet of Things. Internet-of-Things (IoT) Systems*. Springer, Cham, (2018)
3. C. Alexakos, C. Anagnostopoulos, and A. P. Kalogeras, *Integrating IoT to manufacturing processes utilizing semantics*. *Proceedings of IEEE 14th International Conference on Industrial Informatics (INDIN)*, Poitiers, France, July 18-21 (2016)
4. V. Katsourous, C. Koulamas, A. P. Fournaris, C. Emmanouilidis, *Embedded event detection for self-aware and safer assets*, *Proceedings of 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2015*, Paris, France, September 2-4 (2015)
5. J. Lee, B. Bagheri, H. Kao, *Manufacturing Letters*, **3**, 18-23, (2015)
6. H. Lasi, P. Fettke, H.G. Kemper, T. Feld, and M. Hoffmann, *Business and Information Systems Engineering*, **6**, 239-242 (2014)
7. N. Spanoudakis, and P. Moraitis, *Lecture Notes in Artificial Intelligence*, **5925**, 127-142 (2009)

8. V. Chapurlat, D. Diep, A. Kalogeras, and J. Gialelis, *Building and validating a manufacturing ontology to achieve interoperability*, *Enterprise Interoperability II*, Springer London, (2007)
9. D. Diep, C. Alexakos, and T. Wagner, *An Ontology-based Interoperability Framework for Distributed Manufacturing Control*, *Proceedings of 12th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007)*, Patras, Greece, September 25-28 (2007)
10. *Semantic Sensor Network Ontology*, W3C Semantic Sensor Network Incubator Group, <https://www.w3.org/2005/Incubator/ssn/ssnx/ssn> (2005)
11. *Industrial Internet Consortium (IIC), The Industrial Internet of Things Volume G5: Connectivity Framework*, (2017)
12. M. Hankel, and B. Rexroth, *The reference architectural model industrie 4.0 (rami 4.0)*. ZVEI, (2015)
13. S. W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, and A. King, *Industrial internet reference architecture*. Industrial Internet Consortium (IIC), Tech. Rep. (2015)
14. S. W. Lin, B. Murphy, E. Clauer, U. Loewen, R. Neubert, G. Bachmann, M. Pai, M. Hankel, *Architecture Alignment and Interoperability*, *Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper*, (2018)
15. S. Cheshire, and M. Krochmal, *Multicast dns (No. RFC 6762)* (2013)
16. Z. Shelby, K. Hartke, and C. Bormann, *The constrained application protocol (CoAP)*, IETF (2014)
17. U. Hunkeler, H.L. Truong, and A. Stanford-Clark, *MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks*, *Proceedings of 3rd International Conference on Communication systems software and middleware and workshops*, Bangalore, India, January 6-10 (2008)
18. S. Vinoski, *IEEE Internet Computing*, **10**, 90-92 (2006)
19. H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. Lin, *IEEE Access*, **6**, 1706-1717, (2018)