HAL
open science

A Notification-oriented solution for data-intensive enterprise information systems – A cloud manufacturing case

Yongxin Liao, Hervé Panetto, Paulo Cézar Stadzisz, Jean M Simão

To cite this version:

Yongxin Liao, Hervé Panetto, Paulo Cézar Stadzisz, Jean M Simão. A Notification-oriented solution for data-intensive enterprise information systems – A cloud manufacturing case. Enterprise Information Systems, 2018, 12 (8-9), pp.942-959. 10.1080/17517575.2018.1470258 . hal-01791977

HAL Id: hal-01791977

https://hal.science/hal-01791977

Submitted on 15 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Notification-Oriented Solution for Data-Intensive Enterprise Information Systems – A Cloud Manufacturing Case

Yongxin Liao[1], Hervé Panetto[2], Paulo C. Stadzisz[3], Jean M. Simão[3]

[1] *Graduate Program in Production and Systems Engineering (PPGEPS), Polytechnic School, Pontifical Catholic University of Parana (PUCPR), Rua Imaculada Conceição, 1155, Curitiba, Paraná, Brazil. yongxin.liao@pucpr.br*

[2] *Research Centre for Automatic Control (CRAN UMR 7039), University of Lorraine, CNRS, Campus Sciences, BP 70239, Vandœuvre-lès-Nancy, France. herve.panetto@univ-lorraine.fr*

[3] *Graduate Program in Electrical Engineering and Industrial Computer Science (CPGEI), Graduate Program in Applied Computer Science (PPGCA), Federal University of Technology - Paraná (UTFPR), Avenida Sete de Setembro, 3165, Curitiba, Paraná, Brazil. is {jeansimao, stadzisz@utfpr.edu.br*

Abstract: The application of Industrial Internet of Things (IIoT) has brought a series of new challenges to the traditional Enterprise Information Systems (EIS). More and more networked manufacturing entities are combined with data sensing, collection, transmission, and actuation capabilities. EIS **are** now facing large amounts of heterogeneous data. To address this issue, this paper proposes a notification-oriented solution for the data-intensive EIS within a distributed, collaborative, and heterogeneous environment. The solution is then applied in terms of model to a cloud-based smart factory prototype. This work indicates the suitability of the proposition and discusses the existing drawbacks and perspectives that include the manual modelling process and the application on a prototype.

Keywords: data-intensive; enterprise information systems; cloud manufacturing; notification-oriented paradigm; holonic approach; industrial internet of things. ??

1. Introduction

According to the recent Industry 4.0 systematic literature review and research agenda proposal (Liao, Deschamps, et al. 2017), the use of the concept “the fourth industrial revolution” is not something new, once it has been suggested in 1988. It was described as the participation of scientists in the production in order to transform an invention into an innovation (Rostow 1988). In the Hannover Messe 2011, the new concept of “Industrie 4.0” was announced. A set of modern Industry 4.0 implementation recommendations were proposed by the Industry 4.0 Working Group and presented to the German Federal Government (ACATECH 2013). Finally, it became one of the ten future projects in the “High-Tech Strategy 2020” action plan of Germany in March of 2012.

In order to follow this trend, different worldwide countries and regions have carried out various sorts of public policies (Liao et al. 2018). Some examples are the "La Nouvelle France Industrielle" from France (CNI, 2013), the “Made in China 2025” from China (SC 2015), and the "Piano Nazionale Industria 4.0" from Italy (MDSE, 2016). However, the German Industry 4.0 definition received the most popular recognition. It describes this new industrial revolution wave as the technical integration of Cyber Physical Systems (CPS) (Khaitan and McCalley 2015) into manufacturing and logistics, as well as the use of Internet of Things (IoT) (Atzori, Iera, and Morabito 2010) in industrial processes (ACATECH 2013).

As a broad range of network connected, multi-disciplinary, and physically-aware engineered systems, the CPS emphasizes a seamless integration of the cyber world entities into the physical world entities by means of embedded computing technologies (Gunes et al. 2014). In a Cyber Physical Production System (CPPS), each of its smart

manufacturing entity should be able to sense itself and its environment, to collect data, to transmit data, and to react according to those data. Consequently, CPPS generate and consume large amounts of heterogeneous data through the vast endpoints in the so-called informational cloud (Dassisti, et al, 2017).

This situation requires the traditional Enterprise Information Systems (EIS) to embed with new capabilities, such as to exchange information by means of Industrial Internet of Things (IIoT) and to carry out Big Data analytics. Currently, **there** exist some commercial cloud-based IoT platforms (Ren et al. 2015) that provide functions like connectivity, storage, analytics, and visualization in a secure way and at scale. However, the critical issue about how to transform the traditional EIS for adopting those new challenges is remaining open (Zdravković et al, 2016). Moreover, along a production line, the strong inter-relationships between different smart manufacturing entities increase dramatically the complexity to manage their local coherency and their global interoperability (Zdravković et al. 2017).

Therefore, in order to address the above-mentioned data-intensive issue, in the previous work (Liao, Panetto, et al. 2017), the analysis about the new requirements for future EIS was conducted. It has confirmed that the *Self and Environmental Awareness* of smart manufacturing entities is the preliminary need. In this sense, future EIS should be able to accept and process the large amounts of heterogeneous lower level sensory data from distributed or co-located devices (Panetto et al. 2016).

Based on the need of the data-intensive EIS, an ontology-based model-driven pattern was proposed as a potential solution to better model the complex relationships among CPS and to ensure more coherent data exchanges. That solution intends to apply the Notification Oriented Paradigm (NOP) in the manufacturing context from an

ontological viewpoint. It was inspired by the fact that the notification approach is capable of dealing with the control of distributed, collaborative, and heterogeneous systems in the holonic manufacturing context by means of a development paradigm shift (Simão 2005; Simão, Stadzisz, and Morel 2006; Simão and Stadzisz 2009; Simão, Tacla, and Stadzisz 2009; Simão et al. 2012).

As an extension of the previous work, this paper aims to model those data exchanges of the data-intensive EIS via the notification-oriented solution, and then apply it in a cloud-based smart factory prototype. The rest of the paper is organized as follows: Section 2 presents the academic background, namely Holonic Manufacturing Systems and Control, and the extended meta-model of the NOP. Then, Section 3 proposes the NOP data exchange modelling primitives and its detailed application procedures in a cloud-manufacturing scenario. Next, Section 4 introduces a cloud-based smart factory prototype, as the case study test bed and provides corresponding data-exchange modelling examples based on the proposed solution. At the end, Section 5 summarizes this paper and proposes future works.

2. Academic Background: Holonic and Notification approaches

In the domain of manufacturing, the interoperability of Manufacturing Systems (MS) along the life cycle of a product, at different hierarchical levels, in and across industries, has been gradually accepted as one key feature of the fourth industrial revolution (Liao, Ramos, et al. 2017). In this context, the Intelligent MS (IMS) community developed the holonic approach, called Holonic MS (HMS) (Bongaerts, 1998; Wyns, 1999; Simão, 2005; Simão, Tacla, and Stadzisz 2009; Bajic and Chaxel, 1997; Valckenaers et al. 1998; Van Brussel et al. 1998; Van Brussel 1994).

One of the main objectives of the HMS approach is to enable the interoperability and collaboration features of all sort of manufacturing entities in order to improve MS for achieving the nowadays so-called the fourth industrial revolution. Naturally, all these interoperability and collaboration capabilities would be feasible by means of integrating smart computational entities. Therefore, those computational entities should present suitable performance, collaboration agility, and even some emergent behaviour in a comprehensive way. Moreover, those computational entities should achieve these features in the highly distributed environment of manufacturing systems, that represents an additional complexity factor (Simão 2005; Simão, Tacla, and Stadzisz 2009; Deen 2003; Morel and Grabot 2003).

In this given context, a fundamental issue is that the traditional sequential computer science techniques and paradigms are not actually able to fulfil the HMS requirements. These paradigms are Imperative Paradigm (IP) that comprises techniques such as procedural and object-oriented programming and Declarative Paradigm (DP) that comprises techniques such as functional and logic programming. These traditional sequential techniques and paradigms present similar drawbacks, such as the tendency to redundant code and coupled code. This is related to their search orientation (e.g. loop-oriented approaches) over quite passive entities (such as, if-then statements, arrays, and data structures), which tends to create very interdependent pieces of code and code with many temporal and spatial redundancies (Simão and Stadzisz 2009; Simão et al. 2012; Ronszcka et al. 2015; Ronszcka et al. 2017).

Those drawbacks imply on hardness to obtain, for example, suitable performance and uncoupled distribution. Some development and modelling techniques under the principle of module cohesion and decoupling, such as object, aspects, and

event oriented development, applied by very trained developers can even diminish those problems. Nevertheless, they do **not** eliminate them in their essence and, furthermore, they demand highly experimented or formed personal to deal with these paradigm imperfections. Therefore, in order to bridge this gap, the notification-based control was proposed. This concept has evolved and, finally, achieved the form of the nowadays so-called Notification Oriented Paradigm (NOP) (Simão and Stadzisz 2009; Simão et al. 2012).

2.1 Holonic Manufacturing Systems and Holonic Control

The Holonic Manufacturing Systems (HMS) term is emerged from the proper neologism “holon”. The word “holon” was originally proposed by Arthur Koestler in his book <The Ghost in the Machine> (Koestler 1967). It is a combination of the word “holos” (a Greek word that means “whole”) and the suffix “on” (that suggests “particle” or “part”). It refers to the hybrid nature of elements (“whole” and “parts”) in the existing systems, namely the social and biological ones. From the observer’s point of view, on the one hand, holons can be considered as wholes, because they include other existing holons as their parts. On the other hand, they can also be considered as parts, since each holon has the possibility to be a part of the other new holons. Therefore, they are neither made up of a strict hierarchy **nor** a strict heterarchy, but a “holarchy” as defined in the Arthur Koestler’s book (Koestler 1967).

The holonic concept has already been used in some domains where its philosophical nature is applicable (Morel et al. 2003; Clegg 2007; Andrade et al. 2015). In the present context, the legacy manufacturing systems are facing more and more new challenges (e.g. production variability, mass customization, and higher productivity)

with respect to the agility of production. In this context, the HMS were proposed. (Bongaerts 1998; Wyns 1999; Bajic and Chaxel, 1997; Valckenaers et al. 1998; Van Brussel et al. 1998; Van Brussel 1994)

In HMS, the *Product Holons* and *Resource Holons* are envisaged with a kind of intelligence based on their characteristics of autonomy and collaboration. In order to achieve a better agility, the HMS not only requires the composition of various forms of holons, but it also needs an appropriate control mechanism to organize and optimize their collaboration (Bongaerts 1998; Wyns 1999; Simão 2005). Firstly, it was suggested that HMS and its control were implemented by means of Multi-Agent System (MAS) once they have a compatible theoretical background (Deen 2003; Frey et al. 2003; Morel and Grabot 2003). However, in practical terms, both HMS and MAS suffered with the lack of proper computational technique/paradigm to deal with their requirements, in which collaborative behaviour would emerge allowing distributed and smart solutions. Thus, a new solution would be necessary to deal with proper agents or holons of HMS and its essence, the so-called Holonic Control (Simão, 2005; Simão, Tacla, and Stadzisz 2009).

In this context, a Holonic Control (HC) approach was designed by Simão (2005) in order to enable the HMS to achieve an actual "holarchy". This HC has the form of a meta-model in which simple elements would enable the emergence of more complex behaviour and so forth, thereby presenting a sort of fractal property. The proposed HC solution allows the connections between basic holons and promotes their collaborations in a flexible structure, thereby being ruler and coordinator in terms of synergy exploring.

As it can be seen from Figure 1 (a), the HC is concerned to three main kinds of holons, the *Control Rule Holons*, controlled *Resources Holons* and *Product Holons*. The

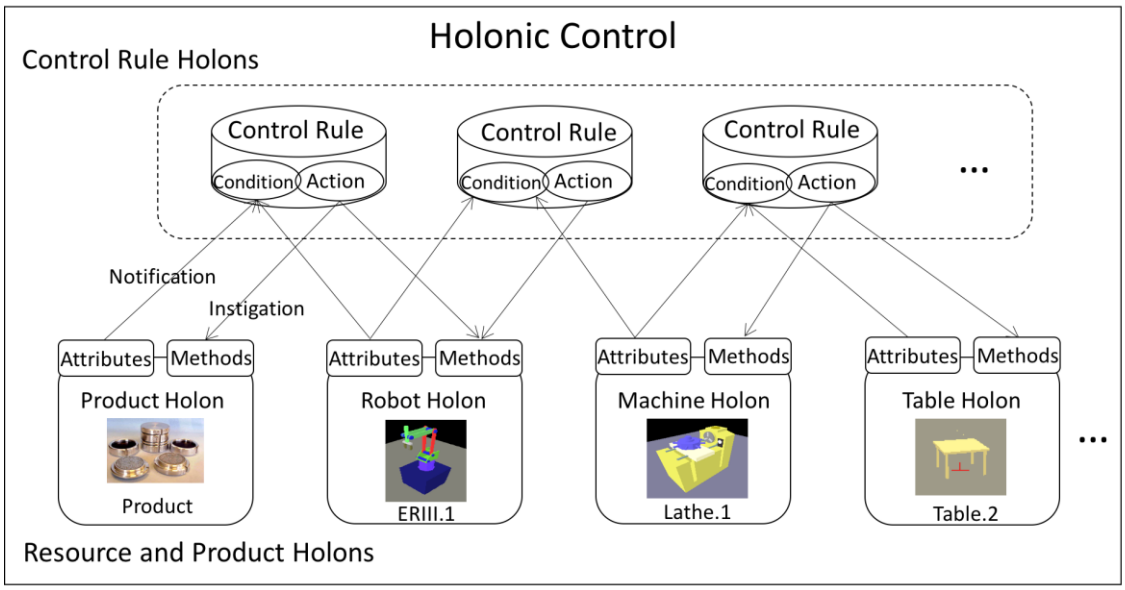
Resource Holons and *Product Holons* (the basic holons in HMS) notify the *Control Rule Holons* about the factual knowledge with respect to their statuses. Each *Control Rule Holon* that is notified will make inference, based on its logical-causal knowledge, about the proper moment to instigate the execution of some control actions. The *Control Rule Holon* is the coordinator of related basic holons, namely *Resource Holons* and *Product Holons*.

The *Control Rule Holon* in HC approach is inspired by the form of logic-causal rules from Rule Base System (RBS). Based on the logic and the knowledge stored in the *Condition* of a *Rule*, it is possible to infer an *Action*. An example, in the form of logic-causal rule, is illustrated in Figure 1(b). If the statuses of the Lathe.1, Table2.1, and ERIII.1 fulfil the *Condition* that is described by *Rule_1.2,* then an *Action* of ERIII.1 Robotic Arm will be instigated. More precisely, the *Transportion.Part Method* in *ERIII.1* will be enabled if the *Lathe.1* is available and contains a finished part, *ERIII.1* Robotic Arm is available, and the *position 2* of *Table.2* is empty.

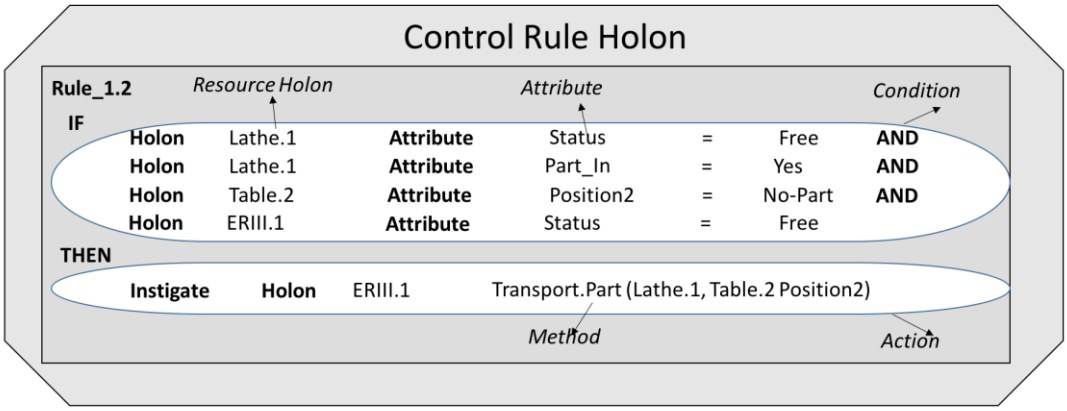
Even though each *Rule* of the HC can be expressed in the RBS form, the inference process is innovative because it is not based on searches, but based on notifications (Simão and Stadzisz, 2009). More specifically, the inference that occurs is following a notification chain and is enabled by a group of Factual-Executional Holons *(Resource Holons* and *Product Holons),* Logic-Causal Holons (*Control Rule Holons*), and also their constituents. The Factual-Executional Holons are composed of *Attributes* and *Methods*, as much as the Logic-Causal Holons are composed of *Conditions*, *Premises*, *Actions*, and *Instigations*.

Figure 1 (c) shows the corresponding example about the inference that is carried out based on the notifications between *Resources Holons*, *Control Rule Holons, and their*

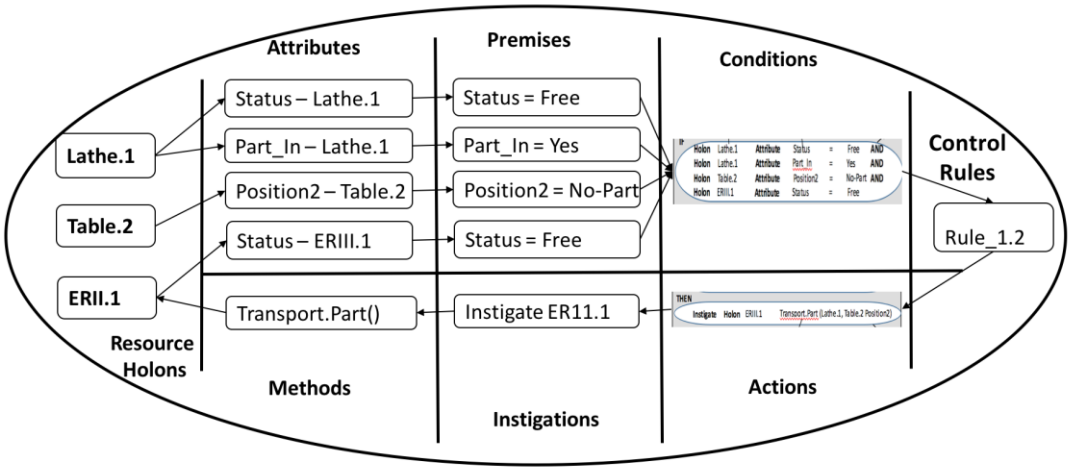
subparts. Those cooperative entities in this system are consistent with the holonic approach that emphasize their nature, as wholes and as parts (Simão and Stadzisz 2009). Moreover, this sort of inference can be used to provide advantages, such as high reactivity, entity decoupling, and the creation of co-operative mechanisms for dealing with determinism and conflict issues (Simão 2005; Simão et al. 2012; Ronszcka et al. 2015; Ronszcka, Ferreira, et al. 2017; Ronszcka, Valença, et al. 2017; Oliveira et al 2018).



(a) Holonic Control



(b) A Control Rule Holon Example



(c) A Notification Example

Figure 1. Holonic Control: the Cooperation between *Resource Holons*, *Product Holons*, and *Control Rule Holons* (Simão 2005)

Subsequently, the research about the HC based on the notification mechanism evolved. First, it was used as a general discrete-control solution (Simão, Tacla, and Stadzisz 2009), and then as a new inference-engine solution (Simão and Stadzisz 2009). Next, it achieved the form of a new programming paradigm after generalization (Banaszewski et al. 2007; Simão and Stadzisz 2008; Banaszewski 2009).

It was designed for software inference according to the precise notifications among the smart, small, and decoupled entities that collaborate together. It makes the composition of both the distributed and not-distributed software, in a way considered easier than existing programming paradigms like the imperative and declarative ones (Simão et al. 2012; Ronszcka et al. 2015; Belmonte et al. 2016; Barreto, Vendramin and Simão 2018; Oliveira et al. 2018).

Moreover, the NOP was also introduced as a new approach for conceiving, constructing, and executing hardware systems (Peters et al. 2012; Linhares, Simão, and Stadzisz 2015; Pordeus et al. 2016; Kerschbaumer el al. 2018). Currently, this solution is understood as a more generic paradigm that not only can be applied to the development and execution of software and even hardware (Wiecheteck 2011; Simão, Stadzisz, and Wiecheteck 2012), but also for the requirements specification and software/system design (Simão et al. 2016; Novaes, Simão and Stadzisz 2018).

2.2 NOP: Meta-Model, Holons and HMS

As the result of the HC evolution, the proposition of the Notification of Oriented Paradigm (NOP) inherits the philosophical nature of the holonic approach, which is concerned to the relativism of wholes and parts. In this sense, Figure 2 presents the proposed NOP meta-model for the generalisation of HC of HMS.

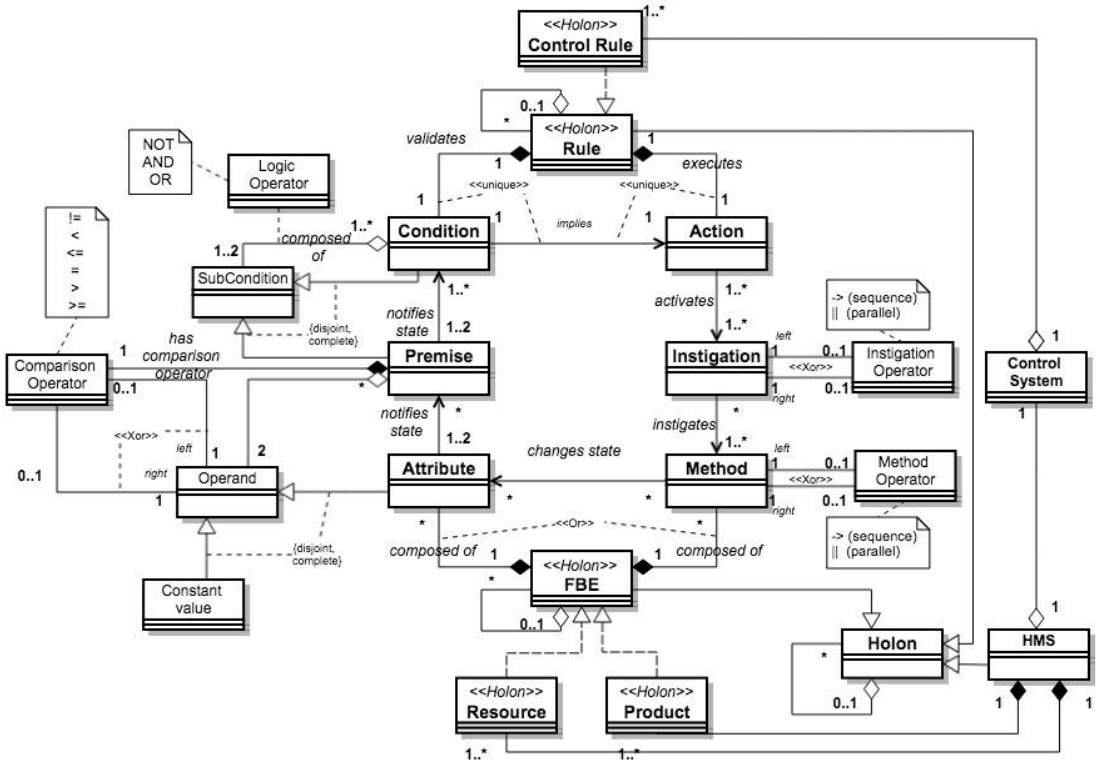


Figure 2. The NOP Meta-Model together with the Concepts of Holon and HMS, extended from (Liao et al, 2017).

In the previous work (Liao, Panetto, et al. 2017), the essential elements of NOP meta-model have been formalized into an ontology-based model-driven pattern, which provides an unambiguously way to model a data transfer. In this paper, the NOP is extended to HMS based on the proposed NOP meta-model ontology, aiming at contributing for the domain of data intensive Enterprise Information Systems (EIS).

As it can be seen from Figure 2, the meta-model of NOP for HMS is adjusted and extended through: (1) the presentation of the key concepts, namely *Holon*, *HMS*, *Product Holon*, *Resource Holon*, and *Control Rule Holon*, and (2) the description of their relationships with the essential elements of NOP.

More specifically, the following concepts, modelled in Figure 2, are taken into account:

- A ***Holon*** can either be considered as a whole that contains some other holons, or be considered as a part that act as one component of another holon. In the manufacturing context, based on holonic NOP, a ***Holon*** is considered as a smart production entity, which should have at least one of the following features:
    (a) Be able to notify the factual knowledge with respect to its status, thereby acting as a factual entity. This feature would be inherited from the pertinent NOP entity class, i.e. *FBE* with *Attribute*.
    (b) Be able to receive instigated commands and perform corresponding responses, thereby acting as an executional entity. This feature would be inherited from the pertinent NOP entity class, i.e. *FBE* with *Method*.
    (c) Be able to receive notified factual knowledge, to make inference based on its logical and causal knowledge, and to establish instigation of control commands, thereby acting as logical-causal entity. This feature would be inherited from the pertinent NOP entity class, i.e. *Rule* with *Condition - Premises* and *Action - Instigations*.

- **HMS** is a specific kind of manufacturing system that is composed of various types of holonic smart production entities, both the physical and cyber ones. There exist three main types of **Holons** in **HMS**: ***Product Holon***, ***Resource Holon*** and ***Control Rule Holon***.
- ***Product Holon*** represents a smart product (Meyer and Holmström 2009). In the proposed solution, *Product Holon* is derived from *FBE* with *Attributes* and *Methods*. Thus, in **HMS**, it should contain at least one of the first two defined ***Holon*** features. In this sense, the notification capability in the feature (a) and the

executional capability in the feature (b) should be explicitly related to the product itself or its environment.

- ***Resource Holon*** represents a smart manufacturing resource. In the prosed solution, ***Resource*** *Holon* is derived from *FBE* with *Attributes* and *Methods*. In ***HMS***, it should also contain at least one of the first two ***Holon*** features. In this sense, the notification capability in the feature (a) and the executional capability in feature (b) should be explicitly related to the manufacturing resource itself or its environment.
- ***Control Rule Holon*** in the Holonic NOP approach takes the form of ***Rules***. It is in charge of the coordination between the ***Resource Holons*** and/or ***Product Holons***. In ***HMS***, it should contain the last ***Holon*** feature, namely the logical and causal inference capability in the feature (c).
- ***Fact Base Element*** **(*FBE*)** is considered as a **Holon** in **HMS**. It allows smart production entities, which can be defined and implemented as either a ***Product Holon*** or a ***Resource Holon***.
- ***Attribute*** is a component of an ***FBE***. It represents one of the ***FBE's*** changeable variables (factual knowledge). Meanwhile it is also in charge of notifying (through a sort of events) its status (when it is changed) to its associated elements, namely ***Premises***.
- ***Method*** is also a component of an ***FBE***. It represents one of the ***FBE's*** functions or executional skills that can be triggered by its associated elements, namely ***Instigations***.

- ***Rule*** is in charge of the logical-causal matter. A ***Rule*** in ***HMS*** can be implemented as a ***Control Rule Holon***, which represents a set of explicit principles that execute the pre-defined ***Action*** when its ***Condition*** is satisfied.
- ***Condition*** is a component of a ***Rule***. It is in charge of the logical organization (*AND*, *OR*, *NOT*) of ***Premises***.
- ***Premise*** is associated to one or more ***Conditions***. It is also linked to either one or two ***Attributes*** for the evaluation of their status. Each ***Premise*** compares the notified status from an ***Attribute*** with a constant or otherwise with the other notified status from another ***Attribute***.
- ***Action*** is a component of a ***Rule***. It is in charge of activating one or more ***Instigations*** and also describes their sequential or parallel relations.
- ***Instigation*** is associated to one or more ***Actions***. It is also linked to one or more ***Methods*** that are able to instigate them.

2.3 NOP and MAS

A Holonic Manufacturing System (HMS) could also be implemented through the Multi-Agent Systems (MAS). In this context, the NOP and MAS can be considered as two synergic approaches, but with different scopes. The MAS use higher-level abstraction concepts (Garcia, Giret, and Botti 2016), whereas the NOP is proposed to deal with basic-level computational objects (as can be seen in Section 2.2) (Simão and Stadzisz 2008; Simão and Stadzisz 2009; Simão et al. 2012; Ronszcka, Ferreira, et al. 2017; Ronszcka, Valença, et al. 2017).

In order to be more specific, the collaborative software agents in MAS are commonly implemented through using the usual programming languages (e.g. Java) that

are based on the imperative and declarative paradigms in terms of execution. These usual paradigms do not provide, at least not in an easily or natural way, internal collaborative entities for each agent (Simão et al. 2012). Therefore, the loop-oriented or search-oriented requests in this kind of approach might cause processing misuse or distribution problems. It is possible to mitigate these issues by using techniques such as objects, aspects, and events, but thereby demanding deep technical and engineering skills (Simão et al. 2012).

By contrast, the NOP programming paradigm is proposed along with an internal collaborative mechanism, through a notification scheme in terms of execution and, meanwhile, in a natural declarative sense in terms of expression. There are only two main types of entities in the NOP: the factual-executional ones and the logic-causal ones, which are designed to deal with the lower-level of computations by emergence and collaborative means (Simão and Stadzisz 2009). This mechanism allows the inference processes to avoid redundancies and coupling, thereby avoiding the misuse of processing and facilitating the parallelism and distributions, beyond other features. Therefore, compared to the other existing programming paradigms, the NOP is more suitable for a MAS implementation once it applies the collaborative principle of entities in more essential levels of computation.

In conclusion, on one hand, a HMS can be directly implemented by the NOP without involving the MAS. On the other hand, the implementation of a HMS through the MAS, by means of usual programming paradigms, would not be able to benefit the features as easy as the ones emerged from the NOP. The main examples of such features are code redundancies avoidance, implicit decoupling, low processing demand, and

implicit parallelism/distribution. In short, beyond HMS, MAS could as well benefit from NOP approach.

3 Proposed Solution

In the fourth industrial revolution era, the Industrial Internet of Things (IIoT) enables the stakeholders to access and use resources for engineering activities via multiple kinds of remote devices and from different internet accessible locations. Meanwhile, it also implies the networked Cyber Physical Production System (CPPS) would generate and consume large amounts of heterogeneous data. In order to adopt this change, the traditional Enterprise Information Systems (EIS) should be able to facilitate the management of the data coming from and going to the whole set of CPPS to become a Data-Intensive EIS.

Hence, how to manage the complex relationships among those smart manufacturing entities and how to ensure more coherence data exchanges among them becomes critical issues. In this section, a cloud manufacturing scenario is introduced in Section 3.1 in order to describe three kinds of data transfer that this paper focuses on. Then, as a potential solution, Section 3.2 explains the holonic NOP data exchange modelling primitives, its graphical notations, and modelling procedures in details.

3.1 Data Exchanges in a Cloud Manufacturing Scenario: a HMS Viewpoint

In the ideal ***HMS***, as it is shown in Figure 3, *the* ***Product Holons*** and ***Resource Holons*** in both the physical and cyber worlds seamlessly collaborate together under the proper organization of the ***Control Rule Holons***. In short, under holonic NOP approach, the ***Control Rules Holons*** (which actually are the ***Rules*** in NOP) receive the real-time data

(***Attributes***) provided by the ***Product Holons*** and ***Resource Holons*** (which actually are the ***FBEs*** in NOP) and coordinate them.

More specifically, the ***Control Rule Holons*** in the form of ***Rules*** (i.e. *Control* ***Rules)*** evaluate the statuses of ***Attributes*** that are described by the ***Premises*** in their pre-defined ***Conditions***. Once the decisional part of ***Rules***, namely the ***Condition***, are satisfied, the execution part of the ***Rules***, namely the ***Actions***, can be carried out. In other words, the sequential or parallel ***Instigations*** in the executed ***Actions*** will trigger the corresponding ***Methods*** in the ***Product Holons*** and ***Resource Holons***.

Actually, the ***Control Rules*** would compose the essence of the Holonic Control System of the HMS. The Holonic Control System would be a sort of holonic EIS dealing with data-intensive sources, namely the Product and Resource Holons or just basic holons, which can comprise IIoT technologies. Naturally, in this scenario, the basic holons of the Data-Intensive EIS can either be concerned in the Cloud or in a local internet accessible data centre. Thus, those IIoT data between holons, which the EIS acquires, are naturally becoming a part of the **HMS**. This becomes the foundation to achieve a better agility of production under IIoT era.

In this context, in order to manage the complex relationships between **Holons** in **HMS** and also ensure more coherence data exchanges between them, besides adopting the NOP Modelling Primitives that have been presented in the previous work (Liao, Panetto, et al. 2017), this paper will extend and explain them into more details, and also focus on applying them in a cloud manufacturing test bed.

In this case, a data transfer from one holon to another is divided into two parts: (1) a notification, which describes the sending part of the data transfer, from a ***Product***

Holon or a ***Resource Holon*** to a **Control** ***Rule Holon***, and (2) an instigation, which describes the receiving part of data transfer, from a ***Control Rule Holon*** to a ***Product Holon*** *or a* ***Resource Holon***. This matter is pertinent to the NOP Modelling Primitives.

Still, before presenting NOP Modelling Primitives, three main kinds of data transfer in **HMS** are listed as follows:

- *Data transfer to enable the collaboration within the physical world*. Namely, the physical collaborations of ***Product Holons*** and/or ***Resource Holons***, under the coordination of ***Control Rule Holons***, at the shop-floor execution level.
- *Data transfer to enable the collaboration within the cyber world*. Namely, the cyber negotiation processes between ***Product Holons*** and/or ***Resource Holons***, under the coordination of ***Control Rule Holons***, at the supervisory management level.
- *Data transfer to enable the collaboration in both the cyber and physical worlds*. It describes the collaboration that includes both above-mentioned kinds.

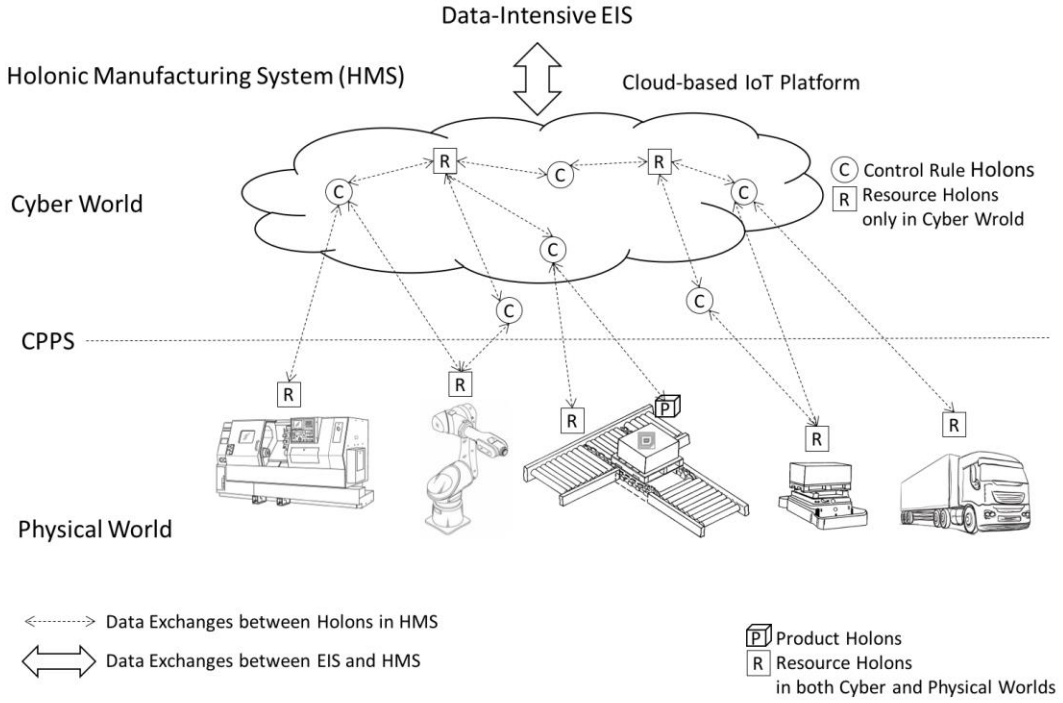


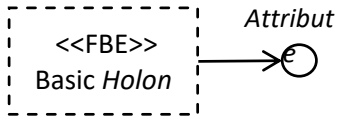
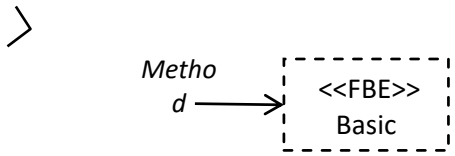
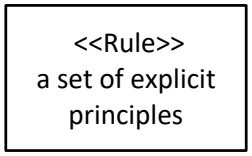
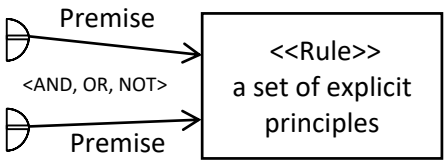
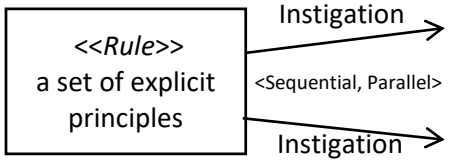
Figure 3. The Data Exchanges in a Cloud Manufacturing Scenario.

3.2 NOP Data Exchange Modelling Primitives and Procedures

In order to better illustrate the NOP data exchange modelling primitives, in this paper, a graphical notation is created for each of its eight-essential elements. Table 1 shows the modelling primitives (column 1), the domain specific meaning for each data transfer in HMS (column 2), and the corresponding graphical notation (column 3).

Table 1. NOP Modelling Primitives, Meanings, and Graphical Notations.

NOP Modelling Primitive	Meaning for Each Data Transfer	Graphical Notations
Fact Base Element (FBE)	The *Source* and *Target* of the data transfer as *Basic Holons* (***Product*** *or* ***Resource Holon***) in **HMS**. It has active entities called ***Attributes*** and reactive entities called ***Methods***	<<FBE>> *Basic Holon*

Attribute	A component of an ***FBE***. It represents the *Variable* inside the *Source* that is changed. It is also an entity responsible for keeping and notifying, in proper time, its status (factual data) via the data transfer.	
Method	A component of an ***FBE***. It represents the *Function* inside the *Target* that can be triggered via the data transfer. It is also an entity responsible for the execution of the *Function*.	
Rule	The *Connectivity* of the data transfer (***Control Rule Holon***) in **HMS**. It describes *a set of explicit principles*. It is composed of a ***Condition*** and an ***Action***.	
Condition	A component of a ***Rule***. It represents the *Pre-condition* that evaluates the *Variables* through a logical organization of its ***Premises***.	
Premise	The *Evaluation Equation* to verify the ***Attribute*** status.	
Action	A component of a ***Rule***. It represents the *Post-condition* for triggering the *Functions* in the parallel or sequential order.	
Instigation	The *Link* to the corresponding *Method* in the *Target*.	

Beyond the NOP Modelling Primitives, it is also needed to define procedures for its using. In this sense, for each kind of data transfer in **HMS**, as listed in Section 3.1, the modelling procedures are introduced as follows:

Step 1. Identification of the data flows among the production entities:

(i) To identify the involved entities in both the cyber and physical worlds.

(ii) To identify the data flows from one entity to another, and their orders.

Step 2. Identification of ***Holons*** (smart manufacturing entities) and the data transfers among them.

(i) To group certain production entities that fulfil at least one of the ***Holon*** features (Section 2.2).

(ii) To identify the *Source* and *Target* of each data transfer. They are ***Product Holons*** and ***Resource Holons*** in **HMS**. Each *Source* should contain at least one *Variable* that can notify ***Control Rule Holons***. Each *Target* should contain at least one *Function* that can be triggered by ***Control Rule Holons***.

(iii) To identify the *Connectivity* between the *Sources* and the *Targets*. It is the ***Control Rule Holon*** in **HMS**. It should contain at least one *Pre-Condition* to evaluate the received *Variable(s)* from the *Source*, and also at least one *Post-Condition* to trigger the linked *Functions* in the *Target*.

Step 3. For each identified *Source* and *Target* in Step 2, an ***FBE*** is created.

Step 4. For each identified *Variable* of the *Source* in Step 2, an ***Attribute*** is created.

Step 5. For each identified *Function* of the *Target* in Step 2, a ***Method*** is created

Step 6. For each identified *Connectivity* in Step 2, a ***Rule*** is created.

Step 7. *Pre-Condition* Analysis and Modelling

(i) To identify the *Evaluation Equation* for each received *Variable*.

(ii) To identify of the *Logical Organization* (AND, OR, NOT) of the *Evaluation Equation(s)*

(iii) For each identified *Evaluation Equation* in Step 7 (i), a ***Premises*** is created

(iv) Combining the identified *Logical Organization* in Step 7 (ii) and the created ***Premises*** in Step 7 (iii), a ***Condition*** is then created.

Step 8. *Post-Condition* Analysis and Modelling

(i) To identify the *Links* to the corresponding *Functions* in the *Target*.

(ii) To identify of the *Execution Order* (Sequential or Parallel) of the *Links*.

(iii) For each identified *Links* in Step 8 (i), an ***Instigation*** is created

(iv) Combining the identified *Execution Order* in Step 8 (ii) and the created ***Instigation*** in Step 8 (iii), an ***Action*** is then created.

Step 9. To merge the same ***FBEs*** from different kinds of data transfer that were previously created.

4 Case Study

In this section, a cloud-based smart factory prototype is presented in Section 4.1 as the case study background. Section 4.2 gives the corresponding modelling examples to show the advantages of the proposed solution.

4.1 Cloud-based Smart Factory Prototype – the Test Bed

In order to show the applicability of the proposed solution in a real cloud manufacturing scenario, the cloud-based smart factory prototype (see Figure 4) developed in the Engineering of Control Automation department of the Pontifical University Catholic of Paraná (PUCPR) is employed as the test bed (Feliciano Filho et al. 2017).

Figure 4 (a) shows the Lego Key Holder that is produced in this smart factory, which aims to achieve all the three preliminary level of intelligence in the smart product classification model (Meyer and Holmström 2009): (1) *Information handing* - The smart product can only manage its own data, but the full control of this product is outside of it. (2) *Intelligence through network* - The RFID tag attached to the smart product acts as an interface to the "intelligence" that exist externally. (3) *Intelligent Item* - The smart product is the smallest intelligent item. In other words, all of its components are not able to be distinguished as individual objects.

As it can be seen from Figure 4(b), this cloud manufacturing scenario starts from the design of customized Lego Key Holders via a mobile (or web) software application connected to IBM Bluemix Cloud. These personalized customer orders will be initially saved or updated (namely, the last-minute changes) into the NoSQL cloud database Cloudant. The communication between the Cloud and the manufacturing resources along the production line, such as Programmable Logic Controller (PLC), MFRC533 RFID Reader and Writer, Conveyor Belts, Automated Guided Vehicle (AGV), and Universal Robots (UR5), occurs through the Raspberry Pi. It serves as a gateway to provide the Internet connectivity. In other words, the exiting manufacturing resources are transformed into IIOT.

Once the Data Access Station 1 (DAS1) is free to accept new customer orders, the Cloud will inform the AGV to transport one empty product container and put it on top of a stand-by Work Piece Carrier (WPC). Then, the corresponding product design of this new customer order will be retrieved from Cloudant and stored into the RFID tag attached to the empty product container. From this moment, this order cannot be

updated any more. The ability to carry out information makes possible the data exchanges between the smart product and the different IIoT manufacturing resources.

Once the WPC moves to the DAS-2, the smart product will communicate with the PLC that controls the Component Dispenser to load the right quantity and right type of Lego bricks into the product container by compressed air. Meanwhile, it also advices the gate (via PLC), next to the conveyor belts, to decide its next manufacturing process: (1) as a semi-finished product to go to the Assembling Workshop, or (2) as a finished product to go to the Warehouse.

In the case of the WPC is moving to the DAS-3 (in the Assembling Workshop), the smart product will communicate with the Tablet to show the customized design of the Lego Key Holder for a human worker to assemble it. After that, when the WPC returns to the DAS-1, the finished smart product (either assembled or not-assembled) will communicate with the AGV to transport itself into one of the three fixed positions inside the Warehouse.

At the end, once the readiness of a mailing box (also with a RFID tag attached) and the readiness of a finished product are both confirmed, at the DAS-4 and DAS-5 respectively, the UR5 robotic arm will pick up the product and put it into the mailing box. Furthermore, during the product manufacturing, customers will be informed the status of their own Lego Key Holders, in real-time, via emails or SMS messages.

Currently, based on the framework of smart factory presented in Figure 4(b), the smart factory prototype has already been designed and partially implemented. In the next sub-section, a real data exchange between the smart product and the IIOT manufacturing resources, from this cloud-based smart factory prototype, is employed as an example.

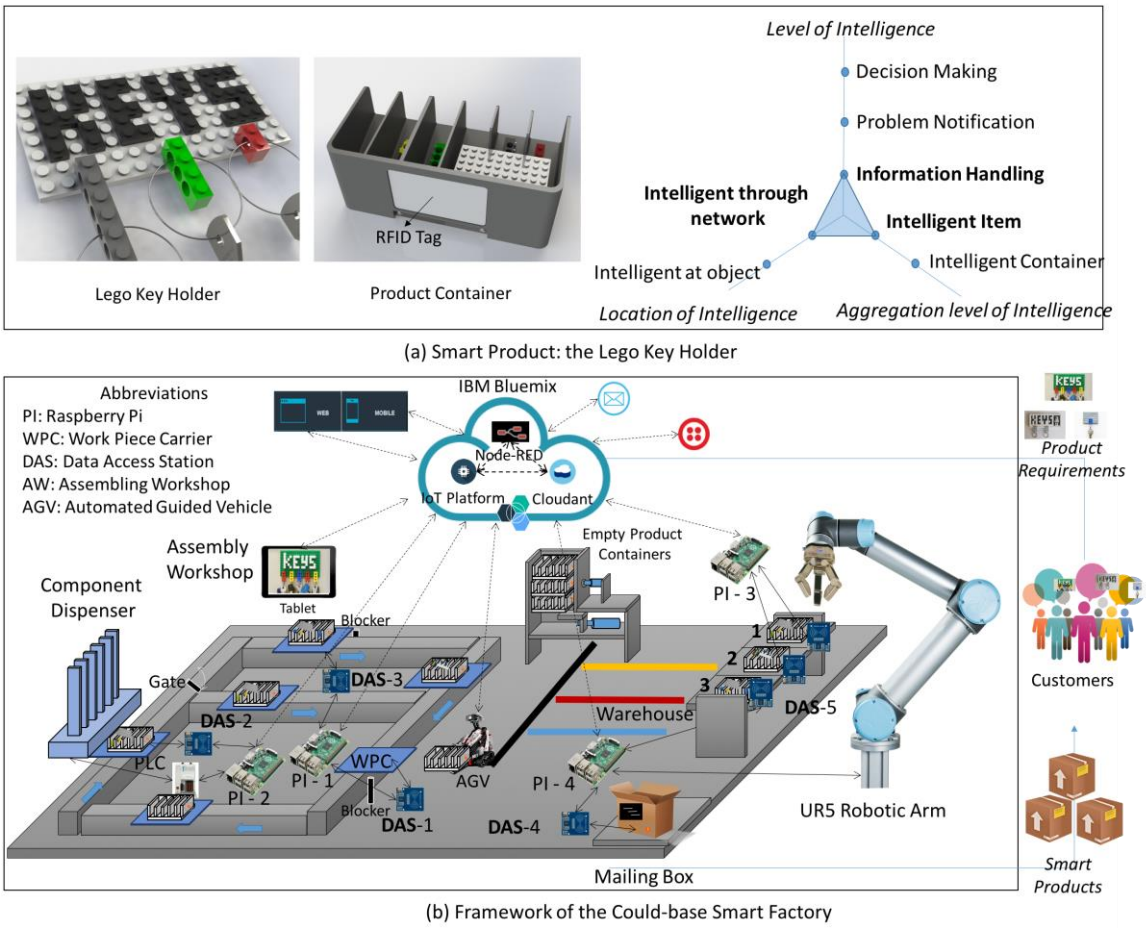


Figure 4. The Smart Product, Cloud-based Smart Factory Framework and its Prototype, upgraded from (Feliciano Filho et al. 2017).

4.2 NOP Data-Exchange Modelling Examples

The finished product boxing process is employed as the application background. Based on the Step 1 defined in Section 3.2, twelve involved production entities, and fifteen data flows from one entity to another are identified in this process. More specifically, as it can be seen from Figure 5, they are explained as follows:

1. MFRC522-W1 in the Warehouse senses the “product_id” from the RFID tag that attached to the Lego Key Holder-1.
2. Raspberry Pi (PI-3) reads the “product_id” from the MFRC522-W1.

3. Raspberry Pi (PI-3) notifies the “product_id” and its “location” in Warehouse to Cloud through Watson IoT Platform (an IoT hub developed by IBM).
4. Watson IoT Platform passes the “product_id” and “location” to Node-RED (a flow-based programming tool for IoT).
5. In Node-RED, if the received “product_id” and “location” are both not empty, then a control command will be sent to Cloudant database to replace the value of “product_status” and “warehouse_no” (of the product with this “product_id”) by the text “in_warehouse” and the received “location” respectively.
6. MFRC522-BS in the Boxing Station senses the “box_id” from the RFID tag that attached to the Mailing Box-1.
7. Raspberry Pi (PI-4) reads the “box_id” from the MFRC522-BS.
8. Raspberry Pi (PI-4) notifies the “box_id” to Cloud through Watson IoT Platform.
9. Watson IoT Platform passes the “box_id” to Node-RED.
10. In Node-RED, if the received “box_id” is not empty, then a control command will be sent to Cloudant database to replace the “box_status” (of the box with this “box_id”) by the text “in_boxing_station”.
11. Cloudant database queries and notifies the values of “box_status”, “product_status”, “warehouse_no”, and “customer_mobile” to Node-RED.
12. In Node-RED, if there exists a box with its “box_status” as “in_boxing_station”, a product with its “product_status” as “in_warehouse”, and the “warehouse_no” and the “customer_mobile” are both not empty, then two parallel control commands will be made. The first

one is in charge of sending a SMS message to the “customer_mobile” through the Twilio SMS Service (a Cloud communications platform).

13. The second one is in charge of passing the “warehouse_no” to Watson IoT Platform
14. Raspberry Pi (PI-4) receives the “warehouse_no” from Cloud through Watson IoT Platform.
15. Raspberry Pi (PI-4) passes the “warehouse_no” to the UR5 Robotic Arm to pick up the Lego Key Holder-1 and put it into the Mailing Box-1.

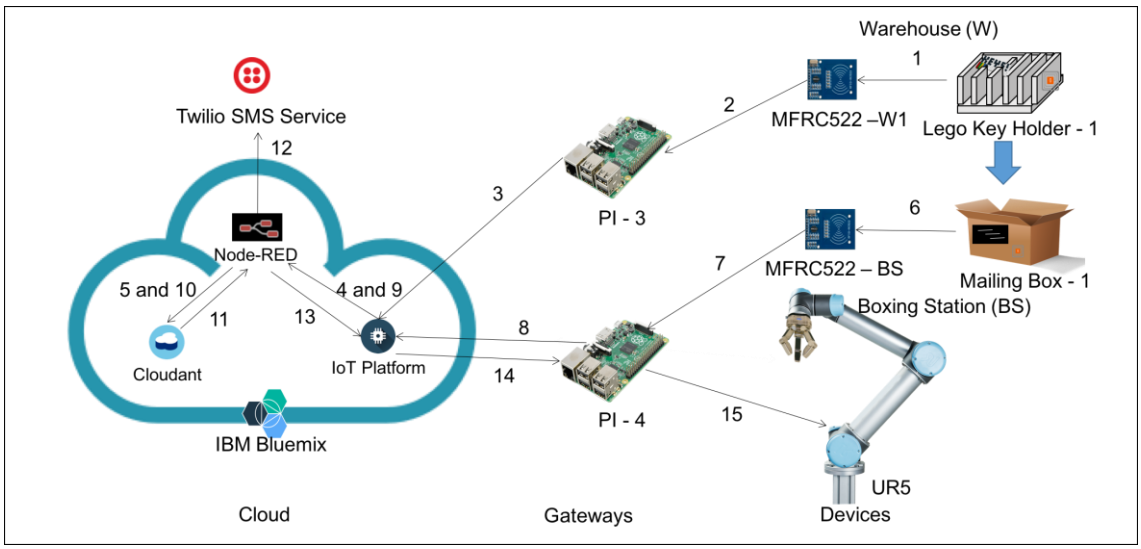


Figure 5. The Involved Production Entities and Corresponding Data Flows

Subsequently, according to the Step 2 in Section 3.2, one *Product Holon*, four *Resource Holons*, three *Control Rule Holons*, and four data transfers between them are identified. More specifically, as it can be seen from Figure 6, they are explained as follows:

- *Product Holon*: it is composed of the Lego Key Holder-1, MFRC522-W1, Raspberry Pi (PI-3), and the data flows among them. The *Holon* feature (a) is used to notify its *Variable* “product_id” and “location”.

- *Resource Holons*:
  - *Mailing Box Holon*: it is composed of the Mailing Box-1, MFRC522-BS, Raspberry Pi (PI-4), and the data flows among them. The *Holon* feature (a) is used to notify its *Variable* “box_id”.
  - *Database Holon*: it is composed of Cloudant Cloud database and its search and storage interfaces (as input and output nodes in Node-RED). The *Holon* feature (a) is used to notify its *Variable* “box_status”, “product_status”, “warehouse_no”, and “customer_mobile”. The *Holon* feature (b) holds the Insert *Function* to be triggered.
  - *Robotic Arm Holon*: It is composed of UR5 Robotic Arm, Raspberry Pi (PI-4), and the data flows between them. The *Holon* feature (b) holds the Boxing *Function* to be triggered.
  - *SMS Service Holon*: It is composed of Twilio Cloud communication platform and its SMS Service interface (as an output node in Node-RED). The *Holon* feature (b) holds the SMS_to *Function* to be triggered.
- *Control Rule Holons*: They are composed of the Watson IoT Platform, Node-RED, and the data flows between them. More importantly, the set of explicit principles are described through Node-RED.
  - *Rule-1*: Its *Pre-condition* is the received “product_id” and “location” are both not empty. Its *Post-condition* is to replace the value of “product_status” and “warehouse_no” in Cloudant (of the product with this “product_id”) by the text “in_warehouse” and the received “location” respectively.

- *Rule-2*: Its *Pre-condition* is the received “box_id” is not empty. Its *Post-condition* is to replace the “box_status” in Cloudant (of the box with this “box_id”) by the text “in_boxing_station”
- *Rule-3*: Its *Pre-condition* is a box with its “box_status” as “in_boxing_station”, a product with its “product_status” as “in_warehouse”, and the “warehouse_no” and the “customer_mobile” are both not empty. Its *Post-condition* is to send a SMS message to “customer_mobile” and to pass the “warehouse_no” to Watson IoT Platform

- *Data Transfers*: The first three of them (DT1, DT2, and DT3) enable the collaboration in both the cyber and physical worlds. The last one (DT4) enables the collaboration in the cyber world.
  - *DT1*: From Product Holon to Database Holon via Rule-1
  - *DT2*: From Mailing Box Holon to Database Holon via Rule-2
  - *DT3*: From Database Holon to Robotic Arm Holon via Rule-3
  - *DT4*: From Database Holon to SMS Service Holon via Rule-3

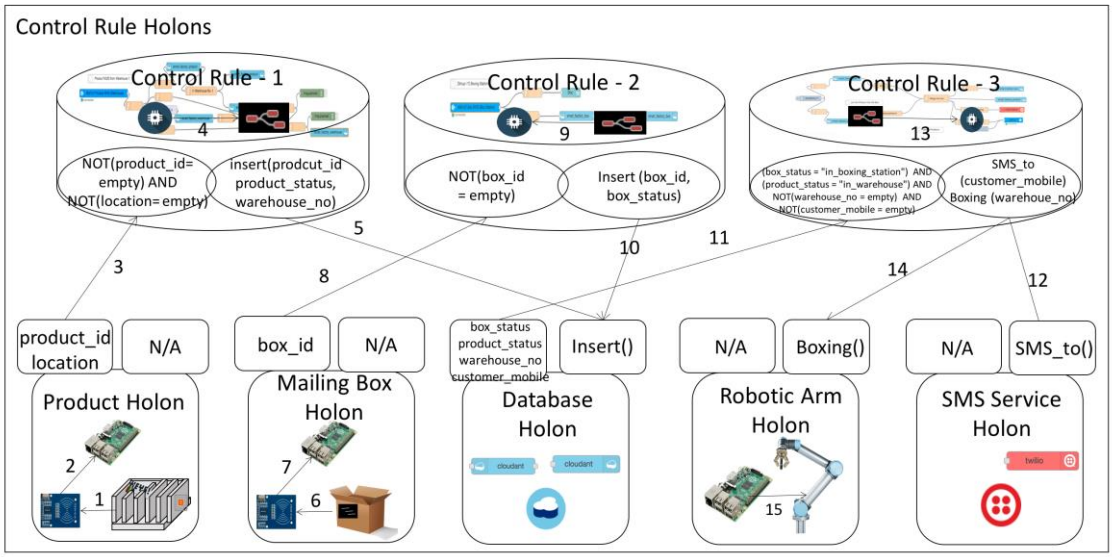


Figure 6. The Identification of the *Product Holons*, *Resource Holons*, *Control Rule Holons*, and their *Data Transfers*

Finally, based on the proposed graphical notations in Table 1 and the steps from 3 to 9 defined in Section 3.2, a NOP data exchange model of the finished product boxing process in the smart factory prototype is illustrated in Figure 7. It can be found that the relationships between all the identified ***Product Holon*** and ***Resource Holons*** are made explicit via the notifications and instigations of the ***Control Rule Holons***. Moreover, these ***Product Holon*** and ***Resource Holons*** also act as the "plug and play" smart manufacturing entities in **HMS**, through their ***Holon*** feature (a) and (b), to ensure more coherent data exchanges between them.

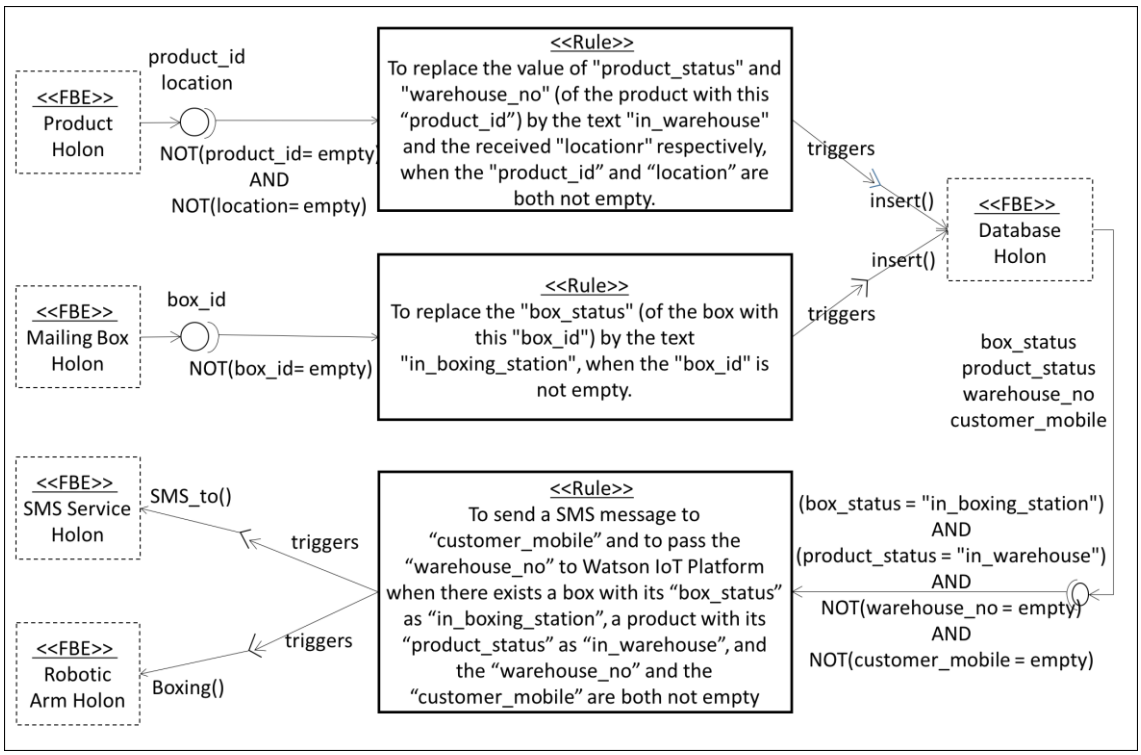


Figure 7. NOP Data Exchange Modelling Example

5 Conclusion

The digitalization of products and manufacturing resources, within the current

manufacturing enterprises, is one of the essential transformations in the fourth industrial revolution. Consequently, the large amounts of heterogeneous data, which are generated and consumed by those digitalized and distributed production entities, require the traditional EIS to adopt to this data-intensive environment.

In this context, the main objective of this study is to propose a potential solution, based on the holonic approach of NOP, to make explicit the complex relationships between smart manufacturing entities and also to ensure more coherence data exchanges among them.

As an extension of the previous work (Liao, Panetto, et al. 2017), this paper first clearly expresses the connections between Holon concepts and NOP elements through a meta-model. Subsequently, the NOP data exchange modelling approach and its procedures are presented in detail to ensure the data-intensive EIS can acquire data from a reliable cloud manufacturing scenario. At the end, in order to demonstrate this concept, the proposed solution is applied on a real data exchange example extracted from a cloud-based smart factory prototype.

Besides the above-mentioned contributions, two main drawbacks are worth to be noted. As first drawback, the modelling process in the case study is currently manually performed. It will be more effective and practical once the automatic extraction of data exchange models from logs and the automatic identification of holons would be completed. As second drawback, the case study was carried out in the smart factory prototype. Even if the prototype result actually can be used to demonstrate the applicability of the NOP data exchange modelling approach and emphasize the possibility of applying the holonic approach to support the cyber and physical

collaborations, an application in real factory would be welcome. These two drawbacks are going to be addressed in the next stage of this research, as a part of the future work.

As a conclusion, in spite of the two supposed drawbacks, once a production entity is integrated with the notification or/and instigation interaction features in a HMS, via the adapter layer of the NOP, it will become consistent with other holons. Actually, this NOP standardisation of interaction and entities helps to achieve the holonic approach main objective, which is interoperability for agile collaboration. In this context, the application of the NOP to support the data-intensive EIS, in either different sizes or market dynamics, is going to confirm the NOP as a promising holonic approach.

Acknowledgment

The authors would like to thank the financial support provided the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), the Pontifical Catholic University of Paraná (*Pontifícia Universidade Católica do Paraná* - PUCPR), and the Federal University of Technology of Paraná (*Universidade Tecnológica Federal do Paraná* - UTFPR), both UTFPR and PUCPR, at Curitiba city, Paraná state, in Brazil.

References

ACATECH (National Academy of Science and Engineering). 2013. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0*. Report , National Academy of Science and Engineering, Frankfurt, Germany.

Andrade, J., J. Ares, R. Garcia, M. A. Martinez, J. Pazos, and S. Suarez. 2015. “A Game Theory Based Approach for Building Holonic Virtual Enterprises.” *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (2): 291–302.

Atzori, Luigi, Antonio Iera, and Giacomo Morabito. 2010. “The Internet of Things: A Survey.” *Computer Networks* 54 (15): 2787–2805.

Bajic, Eddy, Frédéric Chaxel. 1997. "Towards a holon-product oriented management", In *The Proceedings of the 4th IFAC Workshop on Intelligent Manufacturing Systems (IMS 97)*, edited by E. Bajic, 121-126, Seoul, Korea.

Barretto, Wagner Rezende Muniz, Ana Christina Barreiras Kochem Vendramin, Jean Marcelo Simão. 2018. "Notification Oriented Paradigm for Distributed Systems." In The proceeding of the *Computer on the Beach (COTB)*, Florianópolis, Santa Catarina, Brazil.

Banaszewski, Roni Fabio 2009. “Paradigma Orientado a Notificações: Avanços E Comparações [Notification Oriented Paradigm: Advances and Comparisons].” Master diss. Federal University of Technology of Paraná.

Banaszewski, Roni Fabio, Paulo Cézar Stadzisz, Cesar Augusto Tacla, and Jean Marcelo Simão. 2007. “Notification Oriented Paradigm (NOP): A Software Development Approach Based on Artificial Intelligence Concepts.” In *The Proceedings of the 6th Congress of Logic Applied to Technology (LAPTEC 2007)*, 216–222. Santos, SP, Brazil.

Belmonte, Danillo Leal, Robson Ribeiro Linhares, Paulo Cézar Stadzisz, and Jean Marcelo Simão. 2016. “A New Method for Dynamic Balancing of Workload and Scalability in Multicore Systems.” *IEEE Latin America Transactions* 14 (7): 3335–3344. ISSN: 1548-0992. 2016. DOI: 10.1109/TLA.2016.7587639.

Bongaerts, Luc. 1998. “Integration of Scheduling and Control in Holonic Manufacturing System.” PhD diss. Katholieke Universiteit Leuven.

Clegg, Benjamin T. 2007. “Building a Holarchy Using Business Process-Oriented Holonic (PrOH) Modeling.” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 37 (1): 23–40.

CNI (Conseil national de l’industrie). 2013. *The New Face of Industry In France*. Report. Conseil national de l’industrie, Paris, France

Dassisti, Michele, Hervé Panetto, Mario Lezoche, Pasquale Merla, Concetta Semeraro, Antonio Giovannini, Michela Chimienti. 2017. "Industry 4.0 paradigm: The viewpoint of the small and medium enterprises". In *The 7th International Conference on Information Society and Technology (ICIST 2017)*, 50-54. Kopaonik, Serbia.

Deen, S.M .2003. "Agent-Based Manufacturing: Advances in the Holonic Approach". Edited by S.M. Deen, Springer-Verlag.

Feliciano Filho, Marcelo, Yongxin Liao, Eduardo Rocha Loures, and Osiris Canciglieri Junior. 2017. “Self-Aware Smart Product: Systematic Literature Review, Conceptual Design and Prototype Implementation.” In *The Proceedings of the 27th International Conference on Flexible Automation and Intelligent Manufacturing*. Modena, Italy.

Frey, Daniel, Jens Nimis, Heinz Wörn, and Peter Lockemann. 2003. "Benchmarking and robust multi-agent-based production planning and control." *Engineering Applications of Artificial Intelligence* 16(4): 307-320

Garcia, Emilia, Adriana Giret, and Vicente Botti. 2016. “Designing Normative Open Virtual Enterprises.” *Enterprise Information Systems* 10 (3): 303–324.

Gunes, Volkan, Steffen Peter, Tony Givargis, and Frank Vahid. 2014. “A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems.” *KSII Transactions on Internet and Information Systems* 8 (12): 4242–4268.

Kerschbaumer, Ricardo, Robson Ribeiro Linhares, Jean Marcelo Simão, and Paulo Cézar Stadzisz, Carlos R. Erig Lima. 2017. "Notification-Oriented Paradigm to Implement Digital Hardware." *Journal of Circuits Systems and Computers* (accepted)

Khaitan, Siddhartha Kumar, and James D. McCalley. 2015. “Design Techniques and Applications of Cyberphysical Systems: A Survey.” *IEEE Systems Journal* 9 (2): 350–365.

Koestler, Arthur. 1967. *The Ghost in the Machine*. London, UK: Hutchinson.

Liao, Yongxin, Guilherme Louro Brezinski, André Venâncio, Fernando Deschamps, and Eduardo Rocha Loures. 2018. “The Impact of the Fourth Industrial Revolution: A Cross-Country/Region Comparison.” *Production Journal* 28: 1-18

Liao, Yongxin, Fernando Deschamps, Eduardo de Freitas Rocha Loures, and Luiz Felipe Pierin Ramos. 2017. “Past, Present and Future of Industry 4.0 - a Systematic Literature Review and Research Agenda Proposal.” *International Journal of Production Research* 55 (12): 3609–3629.

Liao, Yongxin, Hervé Panetto, Jean Marcelo Simão, and Paulo Cézar Stadzisz. 2017. “Ontology-Based Model-Driven Patterns for Notification-Oriented Data-Intensive Enterprise Information Systems.” In *The Proceedings of the 7th International Conference on Information Society and Technology (ICIST 2017)*, Kopaonik, Serbia. edited by M. Zdravković, Z. Konjović, and M. Trajanović, 148–153. Eventiotic.

Liao, Yongxin, Luiz Felipe Pierin Ramos, Maicon Saturno, Fernando Deschamps, and Eduardo Rocha Loures. 2017. “The Role of Interoperability in The Fourth Industrial Revolution Era.” *IFAC-PapersOnLine* 50(1):12434-12439.

Linhares, Robson Ribeiro, Jean Marcelo Simão, and Paulo Cézar Stadzisz. 2015. “NOCA − A Notification-Oriented Computer Architecture.” *IEEE Latin America Transactions* 13 (5): 1593–1604. ISSN: 1548-0992. DOI: 10.1109/SBESC.2014.28.

Meyer, Gerben G., and Jan Holmström. 2009. “Intelligent Products: A Survey.” *Computers in Industry* 60 (3): 137–148.

MDSE (Ministry of Economic Development). 2016. *Piano Nazionale Industria 4.0*. Report, Ministry of Economic Development, Rome, Italy.

Morel, Gérard, Bernard Grabot. 2003. "Special issue on IMS". *Engineering Applications of Artificial Intelligence* 16 (4)

Morel, Gérard, Hervé Panetto, Marek Zaremba, Frédérique Mayer. 2003. "Manufacturing. Enterprise Control and Management System Engineering: Paradigms and Open Issues." *IFAC Annual Reviews in Control* 27 (2):199-209.

Paulo José Dantas Novaes., Jean Marcelo Simão, and Paulo Cézar Stadzisz. 2018. "Integration between Requirements Modeling and Software Development in the Notification Oriented Paradigm: A Security System Case Study". In *The proceeding of Computer on the Beach (COTB)*, Florianópolis, Santa Catarina, Brazil.

Rodrigo Nunes de Oliveira., Valmir Roth, Alexandre Felippeto Henzen, Jean Marcelo Simão, Percy Nohama, Emilio Carlos Gomes Wille. 2018. "Notification Oriented Paradigm Applied to Ambient Assisted Living Tool". *IEEE Latin America Transactions* 16(2): 647-653. ISSN: 1548-0992. DOI: 10.1109/TLA.2018.8327425

OMB (Office of Management and Budget). (2017). North American Industry Classification System. Report, Office of Management and Budget, Washington, D.C., US.

Panetto, Hervé, Milan Zdravković, Ricardo Jardim-Goncalves, David Romero, J. Cecil, and István Mezgár. 2016. “New Perspectives for the Future Interoperable Enterprise Systems.” *Computers in Industry* 79 (June): 47–63.

Peters, Eduardo, Ricardo Pereira Jasinski, Volnei Antonio Pedroni, and Jean Marcelo Simão. 2012. “A New Hardware Coprocessor for Accelerating Notification-Oriented Applications.” In *2012 International Conference on Field-Programmable Technology*, 257–260. IEEE DOI: 10.1109/FPT.2012.641214.

Pordeus, L. F., Ricardo Kerschbaumer, Robson Ribeiro Linhares, Fernando Augusto de Witt, P. C. Stadzisz, C. R. Erig Lima, and J. M. Simão. 2016. "Notification Oriented Paradigm to Digital Hardware." *Revista SODEBRAS* 11(128): 116-122Ren, Lei, Lin Zhang, Fei Tao, Chun Zhao, Xudong Chai, and Xinpei Zhao. 2015. "Cloud Manufacturing: From Concept to Practice". *Enterprise Information Systems* 9 (2): 186–209.

Ronszcka, Adriano Francisco, Roni Fabio Banaszewski, Robson Ribeiro Linhares, Cesar Augusto Tacla, Paulo Cézar Stadzisz, and Jean Marcelo Simão. 2015. “Notification-Oriented and Rete Network Inference: A Comparative Study.” In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 807–814. IEEE. ISSN: 1548-0992. DOI: 10.1109/TLA.2018.8327425.

Ronszcka, Adriano Francisco, C. A. Ferreira, Paulo Cézar Stadzisz, J. A. Fabro, and Jean Marcelo Simão. 2017a. "Notification Oriented Programming Language and

Compiler". In *The proceeding of the VII Brazilian Symposium on Computing Systems Engineering (SBESC)*. Curitiba, Brazil.

Ronszcka, Adriano Francisco, G. Z. Valença, Robson Ribeiro Linhares, J. A. Fabro, Paulo Cézar Stadzisz, and Jean Marcelo Simão. 2017b. "Notification-Oriented Paradigm Framework 2.0: An Implementation Based On Design Patterns". *Latin America Transactions, IEEE (Revista IEEE America Latina)* 15(11): 2220 - 2231. ISSN: 1548-0992. DOI: 10.1109/TLA.2017.8070430.

Rostow, W.W. 1988. "The 4th Industrial-Revolution And American Society - Some Reflections On The Past For The Future." In *Cooperation And Competition In The Global Economy: Issues And Strategies*, edited by A. Furino, 63–73. San Antonio, TX.

SC (State Council). 2015. *Made in China 2025*. Report, State Council, Beijing, China.

Simão, Jean Marcelo. 2005. "A Contribution to the Development of a HMS Simulation Tool and Proposition of a Meta-Model for Holonic Control." PhD Thesis., Henri Poincaré University (UHP) and Federal University of Technology of Paraná (UTFPR).

Simão, Jean Marcelo, Hervé Panetto, Yongxin Liao, and Paulo Cézar Stadzisz. 2016. "A Notification-Oriented Approach for Systems Requirements Engineering." In *The Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering,* 229–238. Curitiba, Brazil.

Simão, Jean Marcelo, and Paulo Cézar Stadzisz. 2009. "Inference Based on Notifications: A Holonic Metamodel Applied to Control Issues." *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39 (1): 238–250. DOI 10.1109/TSMCA.2008.2006371.

Simão, Jean Marcelo, and Paulo Cézar Stadzisz. 2008. "Paradigma Orientado a Notificações (PON) Uma Técnica de Composição E Execução de Software Orientada

a Notificações [Notification Oriented Paradigm (NOP) — A Notification Oriented Technique to Software Composition and Execution].” Brazilian Patent. INPI Register Number: PI08055181, fied at INPI at 26-11-2008 and at Innovation Agency of UTFPR in 2007.

Simão, Jean Marcelo, Robson Ribeiro Linhares, F. A. Witt, Carlos R. Erig Lima, and Paulo Cézar Stadzisz. 2012. " Paradigma Orientado a Notificações em Hardware Digital [Notification Oriented Paradigm in Digital Hardware]". Brazilian Patent. INPI Provisory Number: BR 1020120264293, field at INPI/Brazil in 2012 and at Innovation Agency of UTFPR in 2012.

Simão, Jean Marcelo, Paulo Cézar Stadzisz, and Gérard Morel. 2006. "Manufacturing Execution Systems for Customised Production." *Journal of Materials Processing Technology* 179: 268-275. DOI: /10.1016/j.jmatprotec.2006.03.064.

Simão, Jean Marcelo, Paulo Cézar Stadzisz, and Luciana Vilas Boas Wiecheteck. 2012. “Perfil UML Para O Paradigma Orientado a Notificações (PON), Perfil UML Para O Paradigma Orientado a Regras (POR), Método de Desenvolvimento Orientado a Notificações (DON) E Método de Desenvolvimento Orientado a Regras (DOR). [UML Profile to the Notificatio UML Profile to the Notification Oriented Paradigm (NOP), UML Profile to the Rule Oriented Paradigm (ROP), Method for Notification Oriented Development (NOD) and Method for Rule Oriented Development (ROD).]” Brazillan Patent. INPI Provisory Number: BR 1020120264307, field at INPI/Brazil in 2012 and at Innovation Agency of UTFPR in 2012.

Simão, Jean Marcelo, Cesar Augusto Tacla, and Paulo Cézar Stadzisz. 2009. “Holonic Control Metamodel.” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39 (5): 1126–1139. DOI 10.1109/TSMCA.2009.2022060.

Simão, Jean Marcelo, Cesar Augusto Tacla, Paulo Cézar Stadzisz, and Roni Fabio Banaszewski. 2012. “Notification Oriented Paradigm (NOP) and Imperative Paradigm: A Comparative Study.” *Journal of Software Engineering and Applications* 5 (6): 402–416. ISSN: 1945-3116. DOI 10.4236/jsea.2012.59083.

Valckenaers, Paul, Hendrik Van Brussel, Jo Wyns, Luc Bongaerts, Patrick Peeters. 1998. "Designing Holonic manufacturing systems." *Robotics and Computer-Integrated Manufacturing* 14 (5-6): 455-464.

Van Brussel, Hendrik, Jo Wyns, Paul Valckenaers, Luc Bongaerts, Patrick Peeters. 1998. "Reference architecture for holonic manufacturing systems: PROSA." *Computers in Industry* 37 (3): 255-274.

Van Brussel, Hendrik. 1994. "Holonic Manufacturing Systems, the vision matching the problem", In The proceeding of the first European Conference on Holonic Manufacturing Systems. Hannover, Germany.

Wiecheteck, Luciana Vilas Boas. 2011. “Método Para Projeto de Software Usando O Paradigma Orientado a Notificações – PON [Software Design Method Using Notification Oriented Paradigm – NOP].” Master diss., Federal University of Technology of Paraná.

Wyns, Jo. 1999. “Reference Architecture For Holonic Manufacturing Systems - the Key to Support Evolution and Reconfiguration.” PhD diss., Katholieke Universiteit Leuven.

Zdravković, Milan, Miroslav Trajanović, João Sarraipa, Ricardo Jardim-Gonçalves, Mario Lezoche, Alexis Aubry, Hervé Panetto. 2016. "Survey of Internet-of-Things platforms". In *The 6th International Conference on Information Society and Techology (ICIST 2016),* 216-220. Kopaonik, Serbia.

Zdravković, Milan, Fernando Luis-Ferreira, Ricardo Jardim-Goncalves, and Miroslav Trajanović. 2017. “On the Formal Definition of the Systems’ Interoperability Capability: An Anthropomorphic Approach.” *Enterprise Information Systems* 11 (3): 389–413.