

# Transitioning From Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges

Luis Ribeiro and Mats Björkman

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-144033>

N.B.: When citing this work, cite the original publication.

Ribeiro, L., Björkman, M., (2017), Transitioning From Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges, *IEEE Systems Journal*, , 1-13. <https://doi.org/10.1109/JSYST.2017.2771139>

Original publication available at:

<https://doi.org/10.1109/JSYST.2017.2771139>

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/index.html>

© 2017 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



# Transitioning from Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges

Luis Ribeiro, *Senior Member, IEEE*, Mats Björkman,

**Abstract**—The concept of Industry 4.0, or the Fourth Industrial Revolution, has the potential for radically increased system reconfigurability and flexibility. At its core, the notion of Cyber-Physical System, as the new generation of embedded systems with advanced artificial intelligence and improved communication capabilities, is seen as the key enabling concept that will render production activities more sustainable. The Cyber-physical conceptualization dramatically reduces the integration effort by virtually eliminating the need, time, and cost for reprogramming. However, there are still important challenges that need to be addressed before one can start to design Cyber-Physical Production Systems consistently. These intertwine and are not as easily solvable as the popular science descriptions may suggest. This work brings them forward and develops a critical comparative analysis between today's automation solutions and their potential cyber-physical counterparts. The analysis considers the technical and conceptual challenges that are included in the process of migrating today's, mostly bespoke, automation solutions to highly modularized, dynamic, and interactive cyber-physical production systems. In this context, the paper considers the interplay between form and function of industrial components, at the light of their cyber-physical formulation. At the same time, it addresses the system-level (de)composability and interaction design challenges that arise from the integration of modular cyber-physical production systems.

**Keywords**—*Cyber-Physical Productions Systems, Agent-based Systems, Holonic Systems, Design, Cyber-Physical Systems, Reconfigurable Manufacturing Systems*

## I. INTRODUCTION

TIME and history have enabled the identification of key disruptive technologies, and their follow-up applications, that where at the core of the so called industrial revolutions. The 4<sup>th</sup> Industrial Revolution (4IR), however, still belongs somehow to the imagination plane whereby society is anticipating unprecedented development at the convergence of emerging concepts and technologies.

Arguably, Cyber-Physical Systems (CPSs) seem to be the main concept supporting the 4IR. CPSs have found many nuanced definitions [1]. Among them, the idea of a logical/cyber representation bound to a physical/mechatronic aggregate and acting as its avatar seems to consistently surface. An automation solution developed upon these principles has a disruptive

advantage in respect to system reconfigurability. This is due to the modular and self-contained nature of cyber-physically formulated equipment and systems that, instead of relying in statically defined and bespoke interconnections, operate in a more open way by considering dynamically establishing and on-demand interactions between the system components. These principles are enablers of more dynamic and sustainable production and business practices whereby industrial equipment can be integrated almost instantly to tackle volatile business opportunities. It can also be promptly disconnected and moved to another plant once the production targets have been fulfilled. The amount of information collected by cyber-physical components could then be used for leasing or renting equipment from a pool with different Quality of Service (QoS) assurances. While a detailed discussion on the potential business models that can be enacted on CPS-like technology is out of the scope of the current work, the interested reader can find more information and the extended rationale in [2].

It is important to notice that the CPS conceptualization requires a data intensive environment as most CPS components will be data collectors and processors. Such direction opens up for deploying the cyber-representation in any computational environment. The intelligence and adaptiveness attributed to this new class of embedded systems therefore not restricted to the computational capabilities of local controllers/devices and can, as well, harness computationally rich cloud environments. This motivation can be seen in the rationale for several international research agendas including, but not limited to [3]: the H2020 Framework Programme for Research and Innovation under the Factories of the Future public-private partnership (EU); the Industrial Internet Consortium, created by AT&T, Cisco Systems, General Electric, IBM and Intel in 2014 (US), the Made in China 2025 initiative (CN) and globally many other initiatives.

The pursue of new industrial automation concepts and solutions, despite being recently fueled by the latter initiatives, and related developments, is also backed up by more than 20 year of multidisciplinary research. These eventually led to several innovative production paradigms and technical contributions namely: Flexible and Reconfigurable Manufacturing Systems (FMS and RMS) [4], [5], [6], Holonic Manufacturing Systems (HMS) [7], [8], Bionic Manufacturing Systems (BMS) [9], Evolvable Assembly Systems (EAS) [10] and, more generally, Multiagent Systems (MAS) [11] and Service Oriented Architectures (SOA) [12], [13], [14]. Most of these contributions have, however, never made it out of the academia or were

Luis Ribeiro (e-mail: luis.ribeiro@liu.se) and Mats Björkman (e-mail: mats.bjorkman@liu.se) are with the Division of Manufacturing Engineering, Department of Management & Engineering, Linköping University, Linköping, SE-58183

applied with important deviations in real operation. Among the main adoption barriers one may mention: the low maturity of the related technologies [15] (particularly considering the cyber-physical interface [16]); limitations with the verification and validation procedures [17] and the lack of integrated architectures that could inform the development, from conceptual stages down to the hardware-on-the-loop implementation level, in a quantitative way [18], [19]. Collectively, these contributions embed the main principles of what would/could be a Cyber-Physical Production System (CPPS) from architectural and technical perspectives but they have never been considered or established in the much required aggregated format.

In addition, the fact remains that the leap from conventional and well known automation technologies to a CPPS is full of uncertainty. Moreover, companies feel lost in the sea of jargon that has suddenly emerged to potentially characterize these new technical systems and that has been opportunistically applied in re-branding existing solutions.

The purpose of this paper is to develop a critical comparative analysis between today's automation solutions and their potential cyber-physical counterparts. The analysis considers the interplay between form and function of industrial components, at the light of their potential CPS formulation. At the same time, it addresses the system-level (de)composability and interaction design challenges. In this context, the paper proceeds with the introduction and definition of the main terminology and background deemed relevant to enact a sound comparative analysis between conventional solutions and CPPSs (section II); the comparative analysis between current automation principles/practices and CPPSs, as well as the clarification of the CPPS conceptual and technical landscape, is detailed in section III and finally, section IV, summarizes the main conclusions and points to future research directions.

## II. TERMINOLOGY AND BACKGROUND

Given the relative novelty of the research in CPPSs it is important to contextualize and position both the terminology and, the related and general architectural directions attributed to the main concepts discussed throughout this paper. First and foremost, the discussion is restricted to the middle levels of the traditional automation stack (Fig. 1) and, therefore, excludes a detailed discussion about the Enterprise Resource Planning (ERP) level as well as the particularities of sensing and actuating upon the physics of specific production processes. It is important to stress, however, that a CPPS encompasses all these dimensions. The following definition for CPPS is considered from [20]:

*Definition 1: "A **Cyber-Physical Production System** is a composition of human resources, production equipment and aggregated products towards which it establishes one or several cyber-physically formulated interaction interfaces. These interfaces are used for monitoring and control of the CPPS operations as well as to tap into the knowledge generated both by the human resources, and the equipment, during the production process as well as knowledge generated by its aggregated products throughout their life-cycle. This internal knowledge is used in different time scales to continuously*

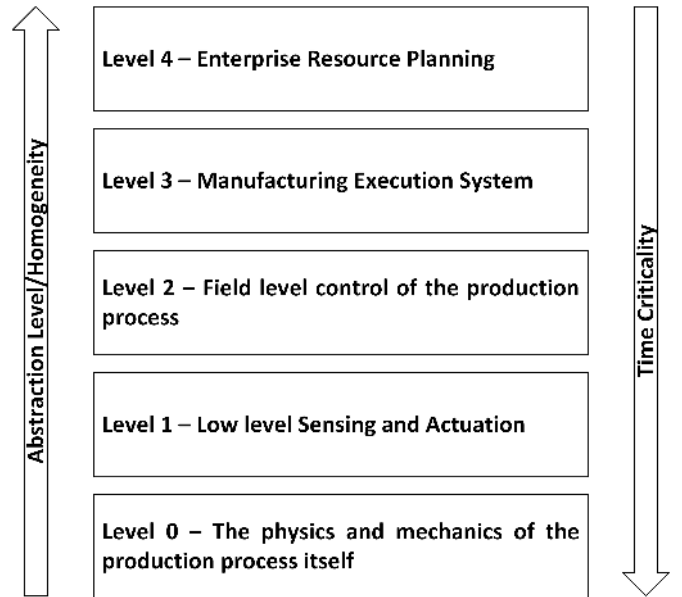


Fig. 1. Conventional automation stack.

*improve operations and to inform the strategic consumptions of capital, raw materials and energy in their many forms. Cyber-physically formulated products will also generate value for external systems, as part of networks of things and services, towards which they maintain interaction interfaces. The outcome of such interactions is external knowledge. Access to it may be offered as a value-added service that can potentially help to further improve CPPS's operations"*

The previous definition implies a compositional approach towards the design and development of CPPSs and a System of Systems (SoS) perspective whereby different entities may dynamic establish relations of different nature in rank, scope and abstraction level. This means that components within a CPPS may follow more integral or modular architectures defined as [21]:

*Definition 2: "(...) **modular architecture** consists of modules, each with one or few distinct functions, connected to each other with a few simple, well-defined interfaces. In the ideal limiting case, all the interactions between the modules occur over these predefined interfaces, and all system behaviour is encompassed by module behaviour and interactions across the defined interfaces."*

*Definition 3: "(...) **integral architecture** contains modules that perform multiple functions and interact over many interfaces (...) in some limiting cases, there are no discernable modules."*

Regardless of the component's internal architectural style it should be seen by other components as a module defined as in [22].

*Definition 4: "A **module** is tightly coupled within and loosely connected to the rest of the system."*

In the scope of the present paper, the main challenges addressed will relate to the integration of industrial equipment and in this respect a Cyber Physical Production Module

(CPPM) can be defined as [20]:

*Definition 5: “A **Cyber-Physical Production Module** is a module (Definition 4) consisting of three logically aggregated entities: an equipment, a controller or computing platform and a cyber representation of this whole. The computing platform may be shared between several cyber-representations if it provides access to the equipment that they represent. The cyber representation contains both the interface and the algorithms that enable the module to interact with other modules without the need for reprogramming it and implements an hardware abstraction layer that decouples the interaction and execution logic from the equipment details.”*

These modules can be developed by considering functional or structural decomposition.

*Definition 6: **Functional Decomposition** is a system decomposition style whereby modules’ boundaries align with available system functions or generalizable roles (order, feeding, supply, etc.) [23], [10].*

*Definition 7: **Structural Decomposition** is a system decomposition style where modules’ boundaries directly overlap with those of the in the physical components [23]. It promotes a physical/mechanical one-to-one relationship and aligns the boundaries of actuation with the ones of the physical body of the abstracted equipment [18].*

Reconfigurability has, for many years, been acknowledged as the characteristic of excellence for mid variety/volume production systems [24]. Even if, nowadays, sustainability is perceived as the excellence paradigm in all activities, reconfigurability is a major pillar of long term sustainability. It enables the usage of shared production resources in different qualities and at distinct sites eventually extending their useful life. In this context, this paper adheres to the following definition of Reconfigurable Manufacturing System (RMS) [6] :

*Definition 8: “A **Reconfigurable Manufacturing System** is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in sudden changes in market or in regulatory requirements.”*

The research in RMS has also set forth the main principles/characteristics that such a systems must have [6], [25], [4]:

- customization - whereby a system is designed to adjust to product families rather than single products or any product;
- convertability - embedded on system design to ease functional changes;
- scalability - to support the increase or reduction of manufacturing resources in a sustainable way;
- modularity - by encapsulating functions to be activated under different production schemes;
- integrability - to support the quick integration of new modules reducing setup and ramp-up times;
- diagnosability - to promptly react to disturbances.

These characteristics can theoretically be satisfied by promoting the following enablers of transformability: universality in system and product design and related technologies; mobility of equipment; scalability along technical, spatial and labor

dimensions; modularity with emphasis in plug and produce models; and compatibility between the systems within the production system.

The research in CPPSs seeks to satisfy these very same requirements by promoting virtually the same enablers. It also does so in a way that is fundamentally different and far more multidisciplinary than today’s automation solutions.

The subsequent section analyses the current practice against potential CPPS solutions considering the most likely trajectories at the light of current research.

### III. A CONCEPTUAL COMPARATIVE ANALYSIS

With the previous background in mind, a comparison between today’s solutions and a CPPS would inevitably have to take into account at least the following points:

- the notion of module, its boundaries and actuation scope - modules in a CPPS will have variations regarding structural and functional complexity and level of aggregation. These variations often occur, by design, in some reference architectures to enlarge the base of systems that they could support.
- the module’s interfaces - the definition of interfaces is of paramount importance in the context of CPPSs as they constrain the set possible interactions within and outside the scope of the system.
- the nature of the control path and module’s interactions - the control path in a CPPS is fundamentally different from conventional automation solutions and several directions have been explored. Some solutions have mainly a conceptual value as their implementation entails a complexity that may not yet be within reach with the available technology.
- the supporting technologies - supporting technologies have been one of the major barriers to the adoption of earlier paradigms. The dominant automation technologies are still unadjusted for CPPS operation while there is a generalized confusion between the computational scope of CPPSs.

#### A. Modules, Boundaries and Actuation Scope

The design of current automation solutions is mostly bespoke. As such, the setup of an automated production system always comprises an extensive and comprehensive programming effort that follows the equipments’ installation.

Production equipment is naturally modular and self-contained. For example, a robot’s design/geometry does not make any assumptions about the state of the environment in which it is going to operate nor about the remaining mechatronic infrastructure it may interact with. It is by default a mechanically and logically decoupled unit. The same could be said about other components that may interact with that robot as part of a specialized station or even the robot’s components themselves.

Individual components, however, rarely add value on their own. They are, instead, part of production processes that make use of one or several of their functions and which require some level of mechanical and logical aggregation. Traditional

automation solutions, particularly at level 2 of the automation stack (Fig. 1), tend to create static and strongly hierarchical logical bounds between all the components (integral design). For example, such solutions rely in hierarchical networks of Programmable Logic Controllers (PLCs) whose programs are custom-designed for that specific (sub)system and redesigned if the (sub)system suffers any changes.

Several consequences result from the latter approach. The integral design generally maximizes system's raw performance but there is a trade-off. Even if the software is contained in one or several controllers that are part of network, the general lack of modularization substantially enlarges its scope of action misaligning it with the one of the physical components. In the event of a failure, and depending on the relative position of the PLC/Program on the network, large sections of the production system can be affected. Such a structure also significantly affects convertability, scalability and integrability. For many years, the ability to reprogram a system has been understood as a central pillar of these three characteristics. While this might have been true at the advent of industrial automation, the complexity of today's systems implies that reprogramming is a time-consuming and error prone activity upon which the ability to reconfigure a system cannot be enacted. The same is valid for the ability to customize.

Flexible Manufacturing Systems have, as a general paradigm, set forth the proposition that multifunctional industrial equipment could solve most customization challenges. Different machine programs could potentially solve the difficulties in adjusting to different products and, for well defined part families, the re-parametrization of existing programs would cater for an effective production without affecting significantly the machine setup times. FMS have been successful for relatively small production volumes but have failed in mass customization scenarios due to performance and cost limitations. Dedicated solutions, on the other hand, normally require high production volumes. In a mass-customization scenario, dedicated equipment would be needed to satisfy the production of a small series within a larger system without disrupting its operations. In such cases, re-parametrization is only part of the equation. The dedicated equipment must be integrated almost instantly while minimizing its introduction's disturbing effect. The ability to move and instantly integrate such equipment in different systems would factor in as a cost justification.

Depending on their design, conventional solutions may be more or less difficult to diagnose. While a pure hierarchical structure should conceptually be easier to analyse, verify and diagnose, due to the scalability needs most automation solutions evolve in time, and practice, to rather unstructured solutions with several "work-arounds" to accommodate the need for change. Such phenomena occurs both at logical and physical levels.

Additionally, conventional solutions are also very much stratified according to the traditional automation stack which means that the complexity of analyzing the overall system is not merely restricted to the field level automation where the programming patterns and system structural organization would arguably be more predictable.

On the other hand, CPPSs, as collections of CPPMs, organize in a rather different way [18]. Instead of considering the bespoke logic boundaries of traditional systems, CPPSs attempt to align them with the value-added functional and structural dimensions of industrial equipment. A CPPM is subsequently developed around them. In this context, a change of function can be enacted by logically reconfiguring a specific equipment, to accommodate a small changes, but also by exchanging CPPMs within the same system or even between systems (system level reconfiguration)

Doing so, however, is not without challenges and aspects of modularity, granularity and (de)composition/aggregation immediately surface. A clear idea about what is a value-added dimension or factor needs to be set as the main rationale around which the CPPSs reference architecture will be developed. Traditional automation system design aims at sheer throughput performance and hence an integral design can be justified. CPPSs' concepts for value-added dimensions or features seem to revolve around reconfigurability and adaptive response. As such, a CPPM must engulf a mechatronic aggregate whose form and functions are positioned at a granularity level not so coarse, that it denotes the same challenges as traditional designs, but at the same time not so small, that its fine grain structure would result in the individualization of non value-added components and a necessary performance penalty.

Previous research has proposed several alternatives for this problem by relying more in physical or logical decomposition and of particular interest is the Holonic formulation of production systems. An Holon, the main building block of an Holonic Manufacturing Systems (HMS), is an entity that is simultaneously a part and a whole. A Holon, as such, does not have, conceptually, any granularity limits.

Since there is no such thing as a dedicated Holonic programming language, most Holonic designs are transposed into agent or service based architectures or a combination of both in articulation with other technologies [26], [27], [28], [29]. A detailed discussion about the conceptual difference between Holons, Agents and Services is off-topic in this contribution and the interested reader can refer to [30], [31] for a comprehensive discussion. It is important, however, to understand the assumptions and decomposition styles that have been considered in previous research. Well known reference architectures, including but not limited to [18], [32], [33], [7], [34], [35] have, under different nomenclatures, defined the smallest functional entity as a Resource Holon or Agent which may be decomposed into itself and coordinated by some higher abstraction entity such as an Order Holon, a Coalition Leader Agent or a Product Agent/Holon, etc. Resources are also often specialized as value-added resources, which abstract transformative processes, and logistic resources, which abstract transportation processes. Collectively, this leaves a relative design freedom to incorporate different structures/aggregates under the control of resource holons/agents and have them as the primary cyber-physical interfaces with the production equipment. Higher order coordinating entities (orders, coalition leaders, etc) can be develop as purely functional units. However, this design direction still raises the possibility of unbounded structural decomposition which, although architec-

turally important is currently limited by several challenges:

- Every layer that needs to be traversed by information before a decision can be enacted on that information results in a performance penalty that arises from that information being processed at different logical levels.
- Unstructured granularity may also cause interactions between components at distinct abstraction levels. The establishment of these parallel and potentially asynchronous communication channels may create inconsistencies in the global behaviour of the system.

While most research in CPPSs has developed in the direction of combining structural and functional decomposition, from Def. 5 the cyber and controlling components still need a well aligned equipment which means that a single reference architecture will surely not fit all cases. This can be illustrated by exploring an additional conceptual example whose main physical components are detailed in the bottom part of Fig. 2. The figure shows, as an illustration, a table size demonstration system that can be commercially purchased for demonstrating different automation solutions. The system is sold as a collection of modules (a, b and c are examples of such modules) which can be re-arranged in different layouts (9 distinct layouts are possible with the three modules considered). The electronics in each module are designed to directly interface with standard 24 VDC industrial signals. Module a) consists of a storage unit with one conveyor belt and two pushing devices that can redirect products to the different storage areas. Module b) consists of a simulated machining unit and one conveyor belt. Module c) consists of an overhead crane, 2 conveyor belts and an inspection station. The layout of the system and its modules is quite representative of today's systems and is generally misaligned with potential CPPS requirements. For example it does not promote the individualization of sub-components such as conveyor belts, machines, smaller stations, etc. This means that for the nine layouts that are theoretically possible nine different bespoke automation solutions would normally be considered.

A CPPS architecture for such system would have to consider not only the current possible combinations but also any future additions. However, the development of CPPMs for this relatively simple system is far from trivial. The CPPS misalignment that was mentioned before does not enable a proper integration between the cyber component of the CPPMs and its physical part in respect to boundaries and actuation scopes. Previous research would encourage distinguishing between logistic and transformative processes/components. The first would expose functions with the semantics of "transport a product from one point to another" while the second would expose function with the semantic "execute whatever value-added operation is required for a product and existing in your operations catalog".

Such modeling approaches have been demonstrated to work, although with limitations, in different prototype implementation using commercial-off-the-shelf industrial equipment. Several application examples can be found at the web page of the IEEE Technical Committee on Industrial Agents (<http://tcia.ieee-ies.org/>).

In this context, conveyor belts, as well as the overhead crane and the pushing units would be individually controlled and abstracted as logistic holons/agents. The machine would be a transformative resource with a set of value-added functions. Products could be abstracted as product holons/agents and would behave as clients of the other agents. Nevertheless, the particular positioning of the sensors that are able to detect the physical products in this system does not allow an individual conveyor to implement a function with the semantics of "move from one point to another". In this case such a function would require the collaboration of multiple logistic holons/agents. The physical system is inducing constraints on the architecture that render the alignment of CPPMs' structure and function with those of the system complex.

One potential solution for this case could be the development of an overseeing logistic agent that would coordinate the lower order agents (Fig. 2). Still, such a solution would also require a methodology to describe how many of such agents would have to be deployed and which routes would they be responsible for. Again, considering the current example one of these overseeing agents could manage the transport actions from a) to b) while another could manage from b) to c). Nevertheless, these would have to synchronize since the conveyor in b) would be a shared resource.

One could also think of a solution based on a single overseeing logistic agent but for sufficiently large systems this solution would come at a computational cost that could render it unusable in real-time. On the opposite extreme, one could consider a fully distributed decision making algorithm that would promote the negotiation among all the involved logistic agents. Many solutions are indeed possible, and some of them have been characterized in the literature, but they rarely reach an applicability level that goes beyond individual prototypes and hence these potential solutions fail to demonstrate their ability to generalize. Such ability is one of the pillars of CPPSs design and one of the main differentiating factors that would set it apart from conventional integral designs. However, it can only be attained if there is an alignment between the system's form and function. This will necessarily be a challenge when transitioning from conventional systems and their automation solution to CPPSs.

The latter does not mean that CPPS design is endlessly complicated, it simply means that it must reach a balance between being fully tailored, as conventional solutions, and trying to cover an overwhelming number of possible cases that would render its complexity unfeasible. It also goes to show that the design of a CPPM and the positioning of its modularity and granularity levels are far from being as straightforward as some approaches suggest. CPPMs should contain generalizable functions and knowledge. As proposed in [16], [27], [28], [29] there will always be controller dependent code that can be integrated as part of an Hardware Abstraction Layer (HAL) which decouples the CPPS's generic decision making, execution and interaction logic from lower order not generalizable details. This is particularly important concerning the integration between levels 1 and 2 of the automation stack (Fig. 1). However, the violation of such principle by introducing potentially generalizable concepts below the HAL,

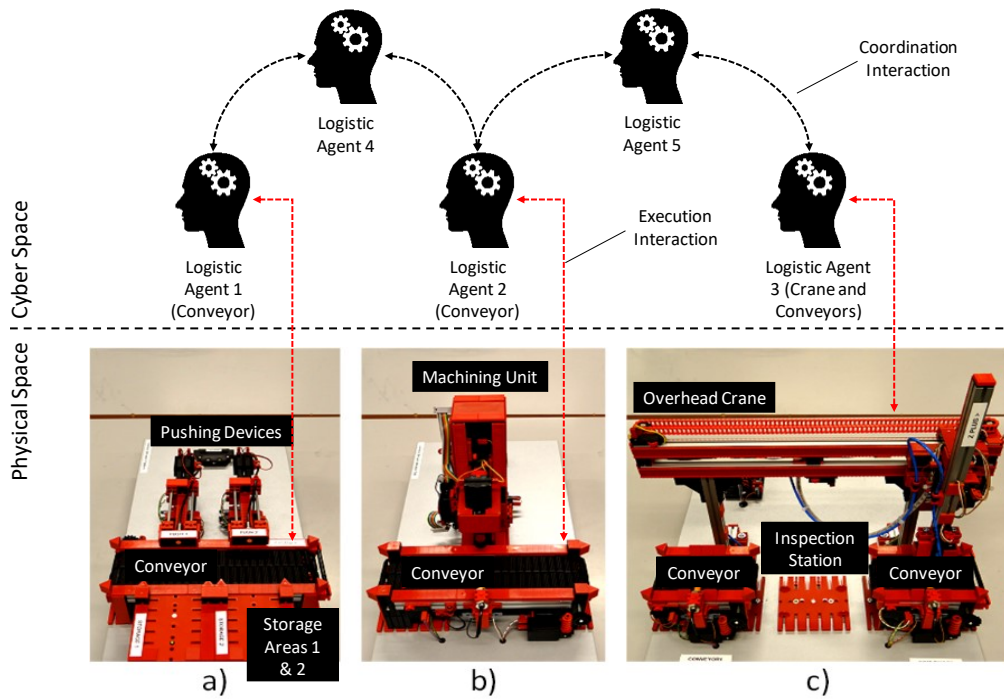


Fig. 2. Potential modularity, scope, and granularity boundaries.

would result in a system with the same fragility of current automation solutions.

A precise understanding of CPPMs boundaries is, therefore, absolutely essential to proceed with the design and implementation of a CPPS. In particular, it influences the design of the module’s mechatronic interfaces upon which the main control strategies and mechanisms are based and which end up constraining the selection of supporting technologies.

**B. Module Interfaces**

Taking into account Def. 5 and the previous discussion it is obvious that a CPPM establishes several interaction interfaces (Fig. 3).

In particular, the CPPM establishes an interface with one or more industrial equipment modules. This composition relation will correspond to a level of mechatronic aggregation that satisfies the reconfigurability requirements of the system, as discussed before. Such composition is also at the core of the cyber-physical formulation of Def. 5. Currently this interface would position itself between levels 1 and 2 of the automation stack (Fig. 1). Individual equipment would be further composed by sensors and actuators yet, the nature of the information processed at that level (intersection of levels 0 and 1) is harder to generalize due to its proximity to the physics of the process. As soon as this information reaches the CPPM it is delegated to the HAL which harmonizes it with the semantics of the CPPS.

The integration of cyber and physical parts of CPPM has historically, and under many different names, been considered in multiple ways. Such developments have also been attached

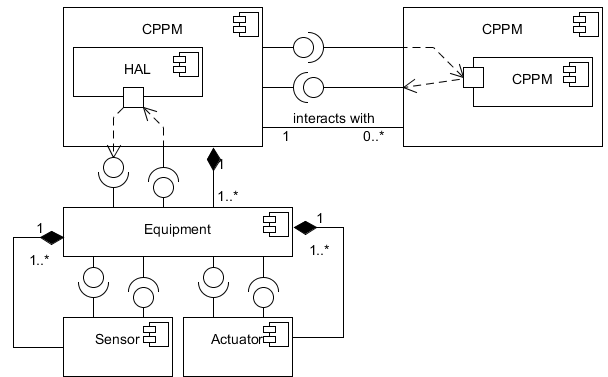


Fig. 3. Cyber-Physical Module Interfaces.

to specific technologies. Currently there is an IEEE Standards Association project, denominated P2660.1 - Recommended Practices on Industrial Agents: Integration of Software Agents and Low Level Automation Functions, that is analyzing and characterizing many different integration practices [36]. The project will produce a set of recommendations for which practice to use according to different criteria taking into account the integration context.

The concept of HAL entails this integration process but also extends it by including a layer that harmonizes the execution semantics of a specific controller or communication protocol with the ones considered in the CPPS. A fully documented example a HAL fulfilling these requirements is detailed in [16].

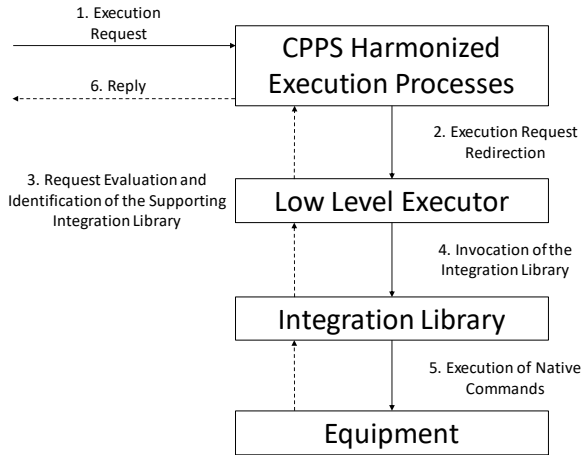


Fig. 4. Simplified Architectural View of an Example HAL.

A simplified architectural view is depicted in Fig. 4.

Interactions between CPPMs in a CPPS should be standardized and described by a common and machine-interpretable model. This common language and set of protocols will most likely differ from the one considered at the physical level where proprietary languages and protocols are normally used. This means that there is an inevitable translation process. Fig. 4 offers, as an example, a simplified view of the work detailed in [37] and later refined in [16]. The HAL within each CPPM is composed by a low level executor that, at any time, can be configured to redirect CPPM-level execution requests to different industrial equipment. Such (re)configuration is executed by means of a configuration file that describes, in the CPPS semantics, the services offered by any given equipment. Furthermore, such file contains a link to an equipment-specific integration library that handles the translation. Fig. 4 broadly describes the whole execution path process.

The key enabling idea is the usage of the generic low level executor together with the integration library. They ensure that as long as the integration library fulfills the execution contract established by the configuration file, then the low level executor can in principle interact with any device that is supported by the library. As such, the rules that govern the control of the system are independent from the specific implementation details at equipment level.

This is in fact one of the main points differentiating CPPS design from conventional automation solutions. Conventional solutions proceed with handling and converting information across many different formats whenever a translation process is required. This would occur between field level controllers using different communication technologies but also across the entire automation stack.

The harmonization potentially offered by creating CPPMs also means that standardized information is available earlier, which simplifies other interactions with additional systems further up in the automation stack. At the same time it will remove additional translation processes and will enlarge the number of levels in the automation stack that a CPPM will

occupy (potentially from 0 to 3). This entails that the nature of the information exchanged through these main interfaces will vary in nature, size and time critically which most likely means the establishment of several communication channels or sub-interfaces.

In this context, it is important to carefully formulate the interfaces between CPPMs acting on a peer to peer basis but also the delegation interfaces for CPPMs holonically formulated. Depending on the control strategy and the control path these interfaces may exist in different number and operate in different qualities. This is another differentiation point. In a traditional automation solution interfaces are immutable in nature and number. One system, subsystem or controller will always connect to another given system, subsystem or controller and exchange the same type of information. On a CPPS, interactions happen in a dynamic way and are not always between the same entities. This means that a CPPM's interfaces will ideally support plug and produce. This subsequently entails that interfaces, by establishing a contract between several interacting parties, should ensure, in a CPPS context, that the module, when plugged-in, receives enough contextual information to be able to operate. Such information could be made available through manual reconfiguration, ideally through automatic configuration but it should never require reprogramming of the module's interfaces or internal algorithms.

### C. Control Strategy and Path

One of the most important decisions in respect to the design of CPPS relates to the nature of the control path between the system components. One of the first classifications appeared in [38] and was later extended in [39]. This classification proposes that control architecture be classified as:

- Hierarchical - whereby lower order components strictly abide by the execution guidelines of the control elements immediately above them.
- Semi-Heterarchical - whereby the system can dynamically adjust its control organization contextually dissolving hierarchies and promoting rank-less forms of organization and later on resuming the hierarchy.
- Fully Heterarchical - whereby the system components interact and are not bound by order or rank.

Hierarchical and more centralized control strategies have been the preferred choice for the implementation of conventional automation solutions. As briefly discussed before, the hierarchical organization of components has the advantage of being more predictable in respect to the flow of information and more responsive under static or predefined interactions. However, if under nominal behaviour a hierarchical control structure can be quite stable, under disturbances it promotes the existence of more centralized points of failure. This has traditionally been handled by ensuring that the computational infrastructure hosting critical systems hierarchically organized introduces an adequate level of redundancy. Such redundancy is generally easier the higher the system is in the automation stack. As programs move from controllers to servers, the redundancy supported by running multiple virtual or real



computing machines can render the execution very robust. However, at field level the failure of one controller has the potential for taking down substantially parts of a system until a manual setup is put in place.

The specter of poor graceful degradation at field level, from conventional solutions, has motivated a substantial amount of research in CPPSs and in the preceding paradigms. It is fundamental to notice that there is not a full overlapping between hierarchical and centralized architectures and that hierarchical solutions, whereby the interaction between the components can be managed in a dynamic way, can actually provide an adequate balance between near optimal response, design simplicity and system adaptiveness and evolution. The most important design feature to attain such a compromise is to endow components in each layer with enough autonomy to enact an adaptive response at the face of disturbances, enabling them to recover from localized changes without having to rely on the components on the layer above. Examples of architectures exploring hierarchical design under highly adaptive systems including plug and produce can be found in [34], [40], [41]

Semi-Heterarchical solutions such as [33] have a tremendous potential for further enhancing the system response under disturbances. These approaches can change the organizational structure of the system to address changing production conditions and particularly disturbances. Unlike a dynamically managed hierarchy, under disturbances the CPPMs should be able to identify the regions of the system affected by the problem and instead of relying only on the CPPMs under the hierarchy they could potentially enlarge their scope of interaction and trigger interactions with other CPPMs at the same level. At that point, a collaborative process based on a newly-formed peer-to-peer interaction would re-synchronize the system's activities until the affected components would be back in operation and the hierarchy could be restored.

While this approach is conceptually simple, its implementation entails important challenges. First and foremost, the CPPMs require assess to more semantic information to decide which part of the hierarchy needs to be dissolved and when it can be restored. Secondly, depending on the broadness of the available information, the new organization could leave the system in a rather sub-optimal operational state. Finally, such suboptimal state could persist even after the hierarchy is restored due to a set of local and cumulative, but globally less desired, decisions. The less hierarchical the control strategy is the more important the definition of CPPMs boundaries and action scope becomes. The risk will mainly result from decision myopia and less controlled emergent behavior.

Alternative solutions have also been proposed in the form of fully Heterarchical solutions. By completely decoupling the control path, a rank-less and orderless organization would maximize the system robustness and the opportunity for accommodating changes. The design complexity of CPPMs reaches the maximum here due to the many conditions and roles that they would have to be prepared to tackle and take. Fully flat solutions are rarely encountered in the literature with wide application in industry. A complete survey and an application example can be found in [42] that shows that most

of the approaches considered along these lines rely in bio-inspired concepts. Typical interaction patterns would include [43]: direct cooperation by pre-establishing cooperation rules, stigmergy by making sure that the interactions are mediated by the environment, direct interactions with access to global information or reinforcement whereby a CPPM would base future decisions on previous experiences. The different patterns will necessarily influence the level of decision myopia as well as the frequency and complexity of the interactions.

More importantly, and unlike natural systems, production systems are characterized by high heterogeneity which often leads to a combinatorial explosion of cases that need to be considered generically. For this reason, even the more advanced control strategies limit the number of possible CPPM types and their sub-interfaces. Natural systems on the other hand are highly homogeneous. Particularly social insects, from where most of the bio-inspired approaches derive from, rely on a high number of alike individuals with a limited variation in the specialization of functions. This ensures that the loss of an individual will never affect the system substantially which is not exactly the case for production systems.

The less hierarchical the control strategy is the more difficult it is to adapt it to current industrial equipment since the metaphors used do not overlap well to what the equipment naturally does nor to the available software models.

#### D. Supporting Technologies

Particularly in less hierarchical proposals, emerging concepts such as: autonomy, evolution, adaptation, adaptedness, self-organization, emergence, scalability, plug-ability, robustness, resilience, etc.; have been applied in different mixes and with different meanings across the technical literature (see [44], [42], [45] for a comprehensive set of definitions).

However, these concepts have also been sometimes elusively applied in connection to the control path classification schemes discussed before. The elusiveness arises when distributed computation is confused in fully or semi-heterarchical architectures with different levels of component autonomy and when the lack of it is attributed to hierarchical designs. There is also confusion between distributed problem solving within and outside the scope of more integral architectures. Fig. 5 attempts to clarify these distinct dimensions.

The main reason why it is so important to position existing development is to understand and be able to decide, from a wide range of available support technologies, the ones that best suit the objectives of a CPPS.

Most modern automation systems are positioned at the intersection between distributed computing and network-based systems. In effect, today's automation systems rely in a network of local controllers and servers to carry out their tasks. They are, nevertheless, statically and hierarchically defined as discussed before. In addition, when optimization is considered in these systems it may explore a multi-computer infrastructure but the optimization algorithm itself will be oblivious to that infrastructure.

Most previous MAS-based contributions have, on the other hand, explored the usage of autonomous entities functionally

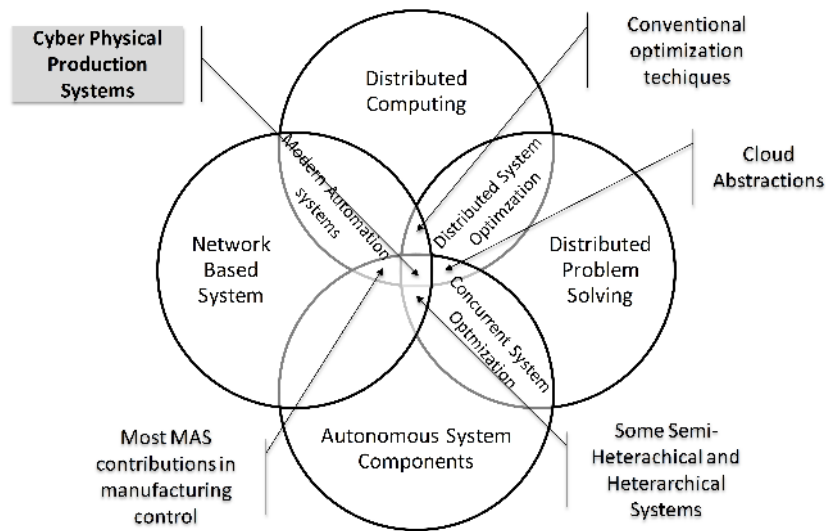


Fig. 5. Dimensions of Distributed, Self-organizing and Bio-inspired designs.

or structurally decomposed to enact a robust and fault tolerant response in networked systems [11], [46], [15]. The more strongly bio-inspired approaches have attempted to do so in the scope of distributed problem solving as well [42], [47], [48], [49], [50], [51], either by embedding the distributed problems solving semantics at the component level (Fully Heterarchical Bio-Inspired Control), or by combining local and global optimization (Semi-Heterarchical Bio-Inspired Control). Cloud abstractions are also increasingly being considered in this scope with the characteristic of hiding the network dimension from the implementation.

Considering the wider expectations for CPPSs, it is anticipated that they will have to harmonize all these dimensions by accepting different computing platforms yet, at the same time, react in an adaptive way to changes along different time frames and structures.

The different actuating time frames present an important challenge. Mainstream automation languages and controllers provide reasonable mechanisms for real time operation. Among these is worth mentioning that recursion and dynamic memory allocation are generally not allowed rendering the size of a program predictable along with a reasonable estimation of the scan time for different programs in devices at field level. Simultaneously, real-time network protocols account for predictable data delivery. Even if complex and impractical, the performance of a conventional automation system would be possible to predict in principle.

However, timing as a design abstraction is not considered in the mainstream programming languages and operating systems [52], [53] that would serve as a support to the main CPPS developments. A partial solution for this problem can be enacted by clearly defining and implementing time based design contracts between real time process control constructs and the cyber part of a CPS. Such solution would enable the clarification and the establishment of the execution expectations for both sides of the design [54]. It would also shed

additional light on the opportunities for hosting the cyber-representation of different objects.

As suggested in [55] the cyber component can generally be located within the computing element of the equipment, or equipment aggregate, being abstracted (i.e. the cyber-component is embedded in the object it represents and follows it). Such an approach promotes modularity and easily enables plug-and-produce of equipment. It has also been the preferred option considered in early MAS architectures where the agent has been seen as the main logical control element. In practice, however, there has historically been a very limited offer of controllers that could natively support agent code. Currently this situation is improving with a wider offer in relatively cheap and computationally powerful controllers but these have had a very limited acceptance in industry.

The former has compelled researchers and practitioners to remove the agent from the controller and instead consider the coupling of both entities through a communication/interaction protocol over a network. Such a vision gained a considerable traction with the improvement of remote operations support but mainly because it is a good fit for the cloud manufacturing abstraction [56].

There will most likely be a third and hybrid model whereby some modules will contain their own cyber-representation (and maybe a remote backup) while others will follow a coupled approach since:

- There seems to be a general trend in the embedded computing community to develop easier to use time-oriented programming languages. Under these conditions, and with the spread of small form factor computing devices, there is the opportunity for progressively replacing the traditional factory control solutions and create CPPMs, at new value-adding granularity levels, based on these new controllers.
- Sufficiently complex and large products will most likely already contain some form of computer. In these cases,

containing the cyber-representation within is relatively straightforward.

- At the same time, in relatively simple products, it may not be technically feasible to introduce a controller. These products could still be monitored by their cyber-representation, probably from a cloud-based platform, using explicit user input or by indirect usage estimation from another component interacting with them.

The ongoing trend in industry is, nevertheless, to run the cyber representation from a cloud environment and limit its level of real-time control over the system by restricting it to the definition of medium to long term goals or reconfiguration actions. This cloud layer then connects to the native control of the plant through some form of industrial middle-ware. The rationale to support this development direction is to:

- Promote the adoption of more intelligent automation setups without the profound changes that the embedded approach would introduce;
- Use the cyber-representation in a computationally powerful environment where it can carry out complex tasks such as optimization and simulation.
- Maintain the native real-time control with a minimum set of harmonizing changes to enable it to connect to the considered middleware.

The authors believe that, in the long term, this strategy will be a transitional solution for the hybrid model discussed before. The management of the cyber representation and the cyber-physical connection itself is a fundamental design aspect that needs to be carefully considered in the scope of CPPSs.

There is an additional technological gap that relates to the availability of tools to assess CPPSs. CPPSs, as envisioned in Def. 1 and Fig. 3, are complex and evolving entities with a high degree of autonomy. This creates important assessment challenges that go beyond the tools and resources available today. The static nature of today's automation systems acts, as discussed before, has a limiting reconfigurability and flexibility factor but facilitates their assessment and the description of their behaviour. Components in these systems are generally self-contained and offer a fixed set of possible behaviours that are only activated under expected circumstances.

CPPPs with all the characteristics that have been attributed to them, as well as all the expectations pending upon them, will operate as open systems, with a variable (in size, structure, time and behaviour) set of components. Such an arrangement makes validation a complex procedure given the complexity of the causality matrix between these components. As such, emergent behaviour is very likely to occur and, in most cases, very difficult to trace back to the root conditions. When emergent behaviour has a positive effect on the system it is important to trace it to reinforce the properties that have produced it. Conversely, when it carries potentially harmful effects, the root causes need to be eliminated. This suggests the use of simulation as a process to describe the emergent behaviour [57] that results from the complex system dynamics.

A complementary strategy is to define, to the best of the designer's capabilities, the boundaries in the decision and configuration spaces that the system and its components are

allowed to explore. In this respect [58] considers three working regions:

- Ideal Region - encapsulates the near optimal operational region and ultimately the target of the embedded optimization algorithms and interaction dynamics;
- Allowed Region - that the system is allowed to explore during transient states as part of ongoing and significant adjusting processes;
- Forbidden Region - that the system should not enter under any conditions.

Simulation and the additional strategies are, however, not without challenges:

- The characterization of the three regions is often difficult and the information scope is directly dependent on all the design dimensions discussed before;
- Simulation processes often offer a limited coverage over the decision and configuration spaces that complex systems may cover;
- Simulation has many facets and while there is existing solid work to simulate at the cyber-physical interface, there are no consolidated methodologies for tackling evolving systems.

An immediate example, in respect to the three previous points, is that, for instance, most discrete event manufacturing simulation tools do not include support for simulation and assessment of plug and produce of equipment nor for self-organizing dynamics. They also generally fail to support the connection of external controlling elements and they assume structurally immutable models during simulation runs.

The situation is slightly better considering the simulation of the cyber-physical interface and the physical part of the system for which several models and practices are already in place. However, this localized simulation offers a limited coverage in entire scope of a CPPS and requires a relatively accurate and complex model of the physical processes.

Still, with all the considered advances in the area, assessment has been traditionally one of the culprits for the poor adoption of the emergent concepts and technologies discussed and given the currently developments will persist as one of the most prominent challenges.

#### IV. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Many of the fundamental ideas and much of the research that constitute the basis of 4IR are not new. There are important similarities and challenges that directly relate to more than 20 years of previous work, as the authors have demonstrated in this paper.

There is currently an important emphasis being placed in technology alone when discussing the base developments for 4IR. Although it is an important dimension of this new development landscape, there are preceding conceptual challenges that will not be solved by technology alone.

There is a general lack of design directions for determining basic aspects of a CPPS development. Among these, the establishment of a solid cyber-physical relation, that connects form to function in a consistent way, across large engineering systems has been at its best elusive. Existing production systems

are, by design, treated as one-off immutable entities. This approach renders them rather inflexible and changes, when required, entail a complex and error prone reprogramming-based reconfiguration process.

CPPS design implies a fundamentally different approach. It also requires a transition strategy whereby traditional solutions are migrated to the 4IR way of conceptualizing systems. Here the technological discussion will play an important role but the conceptual transition will be more dramatic as it entails looking at the system in a different way. When the 4IR becomes effectively established, one should expect to see highly adaptable systems based on fine grained modular components that instantly adjust after being plugged to the system. Until then, there is the opportunity to explore system architectures based on more dynamic hierarchical system organizations.

A necessary first step is the definition of what should be a CPPM in a given production context. This may be different for distinct companies and includes the assessment of the equipment aggregation level that adds value for today's and future production needs. This defines the base line of the reference architecture in respect to system reconfigurability and adaptiveness. The choice of supporting technologies flows from that.

Within a CPPS (middle levels of the automation stack) all the CPPMs will, in the previous context, use the same interaction language and semantics. The usage of hardware abstraction layers will enable the integration of heterogeneous components which will be particularly important in a transitional phase. Between CPPSs (higher levels of the automation) it is, at the moment, difficult to assess how likely they are to be so seamlessly integrated. The current scenario and existing developments show a low level of system interaction standardization beyond specific cases.

The latter means that transitioning from current automation solutions to CPPS is not going to happen overnight. There will have to be a technological and conceptual alignment. This will include the mentioned migration strategy but also the development of a new generation of cyber-physical controllers and modules purposely designed. Without a certain level of cyber-physical harmonization the 4IR will be not more than a succession of costly technological patches.

## REFERENCES

- [1] L. Miclea and T. Sanislav, "About dependability in cyber-physical systems," in *Design & Test Symposium (EWDTS), 2011 9th East-West*. IEEE, 2011, pp. 17–21.
- [2] A. Maffei, "Characterisation of the business models for innovative, non-mature production automation technology," Ph.D. dissertation, KTH Royal Institute of Technology, 2012.
- [3] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.
- [4] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: key to future manufacturing," *Journal of intelligent manufacturing*, vol. 11, no. 4, pp. 403–419, 2000.
- [5] R. M. Setchi and N. Lagos, "Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review," in *Industrial Informatics, 2004. INDIN'04. 2004 2nd IEEE International Conference on*. IEEE, 2004, pp. 529–535.
- [6] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP Annals-Manufacturing Technology*, vol. 48, no. 2, pp. 527–540, 1999.
- [7] 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, pp. 255–274, 1998.
- [8] D. C. McFarlane and S. Bussmann, "Developments in holonic production planning and control," *Production Planning & Control*, vol. 11, no. 6, pp. 522–536, 2000.
- [9] K. Ueda, "A concept for bionic manufacturing systems based on dna-type information," in *Proceedings of the IFIP TC5/WG5. 3 Eight International PROLAMAT Conference on Human Aspects in Computer Integrated Manufacturing*. North-Holland Publishing Co., 1992, pp. 853–863.
- [10] M. Onori, "Evolvable assembly systems: A new paradigm?" in *33rd International Symposium on Robotics*, 2002.
- [11] L. Monostori, J. Váncza, and S. R. Kumara, "Agent-based systems for manufacturing," *CIRP Annals-Manufacturing Technology*, vol. 55, no. 2, pp. 697–720, 2006.
- [12] A. W. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, J. L. Lastra *et al.*, "Industrial cloud-based cyber-physical systems," *The IMC-AESOP Approach*, 2014.
- [13] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 1, pp. 62–70, 2005.
- [14] L. Monostori, "Cyber-physical production systems: roots, expectations and r&d challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014.
- [15] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, 2009.
- [16] L. Ribeiro and P. Linder, "Hardware abstraction layer for java-based agents," in *Proceedings of the Annual Conference of IEEE Industrial Electronics Society*, 2016.
- [17] V. Marik and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 27–35, 2005.
- [18] A. M. Farid and L. Ribeiro, "An axiomatic design of a multiagent reconfigurable mechatronic system architecture," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1142–1155, 2015.
- [19] A. M. Farid, "Measures of reconfigurability and its key characteristics in intelligent manufacturing systems," *Journal of Intelligent Manufacturing*, vol. 1, pp. 1–17, 2014.
- [20] L. Ribeiro, "Cyber-physical production systems design challenges," in *IEEE International Symposium on Industrial Electronics*, 2017.
- [21] E. Crawley, O. De Weck, C. Magee, J. Moses, W. Seering, J. Schindall, D. Wallace, D. Whitney *et al.*, "The influence of architecture in engineering systems (monograph)," *Engineering Systems Monograph*, vol. 1, pp. 1–24, 2004.
- [22] N. Chiriac, K. Hölttä-Otto, D. Lysy, and E. S. Suh, "Level of modularity and different levels of system granularity," *Journal of Mechanical Design*, vol. 133, no. 10, p. 101007, 2011.
- [23] M. Merdan, G. Koppensteiner, A. Zoitl, and B. Favre-Bulle, "Distributed agents architecture applied in assembly domain," in *Proceedings of the Eighth International Symposium on Knowledge and System Sciences*. JAIST Press, 2007.
- [24] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International Journal of Flexible Manufacturing Systems*, vol. 17, no. 4, pp. 261–276, 2005. [Online]. Available: <http://dx.doi.org/10.1007/s10696-006-9028-7>
- [25] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of manufacturing systems*, vol. 29, no. 4, pp. 130–141, 2010.
- [26] L. Wang and A. Haghghi, "Combined strength of holons, agents and function blocks in cyber-physical systems," *Journal of Manufacturing Systems*, vol. 40, pp. 25–34, 2016.

- [27] O. J. L. Orozco and J. L. M. Lastra, "A real-time interface for agent-based control," in *2007 International Symposium on Industrial Embedded Systems*. IEEE, 2007, pp. 49–54.
- [28] I. Hegny, O. Hummer, A. Zoitl, G. Koppensteiner, and M. Merdan, "Integrating software agents and iec 61499 realtime control for reconfigurable distributed manufacturing systems," in *2008 International Symposium on Industrial Embedded Systems*. IEEE, 2008, pp. 249–252.
- [29] P. Vrba, P. Tichý, V. Mařík, K. H. Hall, R. J. Staron, F. P. Maturana, and P. Kadera, "Rockwell automation's holonic and multiagent control systems compendium," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 1, pp. 14–30, 2011.
- [30] A. Giret and V. Botti, "Holons and agents," *Journal of Intelligent Manufacturing*, vol. 15, no. 5, pp. 645–659, 2004.
- [31] L. Ribeiro, J. Barata, and P. Mendes, "Mas and soa: complementary automation paradigms," in *Innovation in manufacturing networks*. Springer, 2008, pp. 259–268.
- [32] P. Leitão and F. Restivo, "Adacor: A holonic architecture for agile and adaptive manufacturing control," *Computers in industry*, vol. 57, no. 2, pp. 121–130, 2006.
- [33] P. Leitão, A. W. Colombo, and F. Restivo, "A formal specification approach for holonic control systems: the adacor case," *International journal of manufacturing technology and management*, vol. 8, no. 1-3, pp. 37–57, 2006.
- [34] J. Barata and L. M. Camarinha-Matos, "Coalitions of manufacturing components for shop floor agility-the cobasa architecture," *International journal of networking and virtual organisations*, vol. 2, no. 1, pp. 50–77, 2003.
- [35] D. McFarlane, V. Giannikas, A. C. Wong, and M. Harrison, "Product intelligence in industrial control: Theory and practice," *Annual Reviews in Control*, vol. 37, no. 1, pp. 69–88, 2013.
- [36] IEEE Project P2660.1 – Recommended Practices on Industrial Agents: Integration of Software Agents and Low Level Automation Functions. [Online]. Available: <https://standards.ieee.org/develop/project/2660.1.html>
- [37] L. Ribeiro and J. Barata, "Deployment of multiagent mechatronic systems," in *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*. Springer, 2013, pp. 71–82.
- [38] D. M. Dilts, N. P. Boyd, and H. Whorms, "The evolution of control architectures for automated manufacturing systems," *Journal of manufacturing systems*, vol. 10, no. 1, pp. 79–93, 1991.
- [39] D. Trentesaux, "Distributed control of production systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 971–978, 2009.
- [40] J. Barata, L. Camarinha-Matos, and G. Cândido, "A multiagent-based control system applied to an educational shop floor," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 5, pp. 597–605, 2008.
- [41] L. Ribeiro, J. Barata, M. Onori, and J. Hoos, *Industrial Agents: Emerging Applications of Software Agents in Industry*. Morgan Kaufmann, 2015, ch. Industrial agents for the fast deployment of evolvable assembly systems, p. 301.
- [42] J. Dias-Ferreira, L. Ribeiro, H. Akillioglu, P. Neves, and M. Onori, "Biosoarm: a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors," *Journal of Intelligent Manufacturing*, vol. 1, pp. 1–24, 2016.
- [43] G. D. M. Serugendo, M. P. Gleizes, and A. Karageorgos, "Self-organisation and emergence in mas: An overview," *Informatica (Slovenia)*, vol. 30, no. 1, pp. 45–54, 2006.
- [44] A. B. Urken, T. M. Schuck *et al.*, "Designing evolvable systems in a framework of robust, resilient and sustainable engineering analysis," *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 553–562, 2012.
- [45] T. De Wolf and T. Holvoet, "Emergence and self-organisation: a statement of similarities and differences," *Engineering Self-Organising Systems*, vol. 3464, pp. 1–15, 2004.
- [46] W. Shen and D. H. Norrie, "Agent-based systems for intelligent manufacturing: a state-of-the-art survey," *Knowledge and information systems*, vol. 1, no. 2, pp. 129–156, 1999.
- [47] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Improving the adacor2 supervisor holon scheduling mechanism with genetic algorithms," in *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON NUMERICAL ANALYSIS AND APPLIED MATHEMATICS 2014 (ICNAAM-2014)*, vol. 1648, no. 1. AIP Publishing, 2015, p. 140008.
- [48] P. Leitão, J. Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 5, pp. 934–944, 2012.
- [49] C. Pach, T. Berger, Y. Sallez, T. Bonte, E. Adam, and D. Trentesaux, "Reactive and energy-aware scheduling of flexible manufacturing systems using potential fields," *Computers in Industry*, vol. 65, no. 3, pp. 434–448, 2014.
- [50] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, "Orca-fms: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," *Computers in Industry*, vol. 65, no. 4, pp. 706–720, 2014.
- [51] P. Verstraete, B. Saint Germain, P. Valckenaers, H. Van Brussel, J. Belle, and H. Hadeli, "Engineering manufacturing control systems using prosa and delegate mas," *International Journal of Agent-Oriented Software Engineering*, vol. 2, no. 1, pp. 62–89, 2008.
- [52] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.
- [53] J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia, and J. Zou, "Distributed real-time software for cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 45–59, 2012.
- [54] P. Derler, E. A. Lee, S. Tripakis, and M. Törngren, "Cyber-physical system design contracts," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*. ACM, 2013, pp. 109–118.
- [55] L. Ribeiro, *Industrial Agents: Emerging Applications of Software Agents in Industry*. Morgan Kaufmann, 2015, ch. The design, deployment, and assessment of industrial agent systems, pp. 45–63.
- [56] A. Verl, A. Lechler, and J. Schlechtendahl, "Glocalized cyber physical production systems," *Production Engineering*, vol. 6, no. 6, pp. 643–649, 2012.
- [57] M. A. Bedau, "Is weak emergence just in the mind?" *Minds and Machines*, vol. 18, no. 4, pp. 443–459, 2008.
- [58] R. Frei and G. Di Marzo Serugendo, "Concepts in complexity engineering," *International Journal of Bio-Inspired Computation*, vol. 3, no. 2, pp. 123–139, 2011.



**Luis Ribeiro** (M'07-SM'17) received the M.Sc., in Electrical and Computer Engineering, in 2007 and the Ph.D., in Electrical and Computer Engineering with specialization in Robotics and Computer Integrated Manufacturing, in 2012 from the Universidade Nova de Lisboa (UNL), Lisbon, Portugal. He received a Swedish "Docent" degree in Manufacturing Engineering, in 2016, from Linköping University (LiU), Sweden.

He is currently an Associate Professor at the Division of Manufacturing Engineering, Department of Management and Engineering from Linköping University, Linköping, Sweden. His research interests include the design, development and implementation of the next generation of production systems and its associated challenges namely: dynamic product routing under plug and produce conditions, runtime deployment of components, reconfigurability measurement, simulation and simulation in the loop, distributed system-level monitoring and diagnosis, advanced shop-floor logistics (focusing in product flow) and intelligent and self-organizing industrial systems.



**Mats Björkman** received his PhD degree in Assembly Technology from Linköping University (LiU), Linköping, Sweden in 1990.

He is currently a Professor and, since 2012, the head of the division of Manufacturing Engineering at the Department of Management and Engineering from Linköping University, Linköping, Sweden and from 2007 to 2011 he has been the Deputy Head of the Department of Management and Engineering. His research has a focus on development, design and operation of flexible manufacturing systems and equipment with a high degree of automation. The research has also expanded to areas such as the interface and integration between manufacturing and product design, including design science. He has been involved in and/or being project leader/coordinator for a large number of international and national research projects in these fields.