Empowering a Cyber-Physical System for a Modular Conveyor System with Self-organization

José Barbosa[1,3], Paulo Leitão[1,2], Joy Teixeira[1],

[1] Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança, Portugal,
Email: {jbarbosa, pleitao}@ipb.pt
[2] LIACC – Artificial Intelligence and Computer Science Laboratory, Rua Campo Alegre 1021, 4169-007 Porto, Portugal
[3] INESC-TEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

Abstract. The Industry 4.0 advent, advocating the digitalization and transformation of current production systems towards the factories of future, is introducing significant social and technological challenges. Cyber-physical systems (CPS) can be used to realize these Industry 4.0 compliant systems, integrating several emergent technologies, such as Internet of Things, Big data, cloud computing and multi-agent systems. The paper analyses the advantages of using biological inspiration to empower CPS, and particularly those developed using distributed and intelligent paradigms such as multi-agent systems technology. For this purpose, the self-organization capability, as one of the main drivers in this industrial revolution, is analysed, and the way to translate it to solve complex industrial engineering problems is discussed. Its applicability is illustrated by building a self-organized cyber-physical conveyor system build up with different individual modular and intelligent transfer modules.

Keywords: Cyber-physical systems, Multi-agent systems, Self-organization.

1 Introduction

Manufacturing remains a key force to drive the world's economic growth. In particular, the manufacturing sector employed 31 million persons in EU-27's in 2009, and have generated EUR 5 812 billion of turnover and EUR 1 400 billion of value added [6]. In the last years, industrial manufacturing companies are facing strong pressures by customers that demand more customized and high-quality products [7], which requires the implementation of more flexible, reconfigurable and responsiveness production systems to maintain their competitiveness levels.

The adoption of Industrie 4.0 principles [11], known as the 4$^{\text{th}}$ industrial revolution and characterized by the digitization of traditional factories, allows the companies to be more competitive facing the continuous pressure imposed by global markets and demanding customers. An important remark is that this revolution will bring significant benefits with only about 40 to 50 percent of replacement of equipment [3]. The use of Cyber-physical systems (CPS) [1] can be seen as a way the backbone platform to implement the Industry 4.0 vision, complemented with disruptive technologies, which according to the the McKinsey's report [3] can be grouped into four clusters: i) data, computational power and connectivity (e.g., big data, Internet of Things (IoT), Machine-to-Machine (M2M) and cloud technologies), ii) analytics and intelligence (e.g., data mining, artificial intelligence and machine learning), iii) human machine interaction (e.g., virtual and augmented reality), and iv) digital to physical conversion (e.g., additive manufacturing and advanced collaborative robotics).

Multi-agent systems (MAS) [8, 14] is a proper technology to implement distributed intelligence in CPS solutions, complemented with other emergent technologies, such as Service-oriented Architectures, cloud computing and Big data analytics. MAS allows the development of large-scale complex engineering problems by decentralizing the control functions over distributed and intelligent software agents, that cooperate together to achieve the system goals. This kind of systems needs to be modular, flexible, robust and scalable, but also be aware of reconfiguration, adaptation and evolution in a very fast, automatic and self-manner.

This challenge can be addressed by enhancing these systems with simple but powerful biological mechanisms, which are working in nature by millions of years, to be more responsiveness and agile to emergence. Self-organization is one powerful concept that can be found in several domains, such as biology (e.g., the ants foraging and birds flocking), chemistry (e.g., the Belousov-Zhabotinsky reaction), physics (e.g., the 2^{nd} thermodynamics law) and social organization (e.g., traffic and pedestrian walk in crowed environments). Basically, self-organization is a process of evolution where the development of emergent and novel structures takes place primarily through the system itself, and normally triggered by internal forces. The challenge is to understand the principles of this concept and to translate them, case-by-case, to solve the engineering problems (note that in some situations, the application of these methods didn't reach the desired results since the concepts were simply copied).

Having this in mind, this paper discusses the benefits of enhancing agent-based systems with self-organization capabilities. In particular, the deployment of a self-organized CPS for a modular conveyor system will be described to show that the plugability and the dynamic system reconfiguration can be performed automatically and on-the-fly, i.e. without the need to stop, re-program and re-start the system. This issue is completely aligned with the requirements associated to the Industry 4.0 initiative.

The rest of the paper is organized as follows: Section 2 overviews the upcoming challenges of CPS and introduces self-organization as an aggregation factor for the CPS implementation. Section 3 describes the case study based on modular Fischertechniks conveyors and Section 4 describes the engineering of the self-organized CPS. Section 5 presents the self-organization mechanisms aiming the plugability and reconfiguration of conveyors on the fly. At last, Section 6 rounds up the paper with the conclusions.

2 Self-Organization in Cyber-Physical Systems

2.1 Overview of the Cyber-Physical System Concept

CPS is a paradigm, initially introduced in 2006 by a working group composed of experts from the USA and European Union, which refers a network of interacting computational and physical devices [12], suitable to build complex and large-scale systems. This paradigm advocates the co-existence of cyber and physical elements [15] with a common goal to build complex systems following the principles of system of systems, which simplifies their understanding, design and deployment. CPS are being applied to different domains, namely smart industrial production, smart electrical grids, smart logistics, smart traffic control, smart e-health, and smart cities and buildings.

As stated by [15], the adoption of CPS in industrial environments is faced by several challenges, namely:

- *CPS Capabilities*, comprising amongst others the modularization and servification of CPS systems and the consideration of advanced (big) data analytics.
- *CPS Management*, including security and trust in the management of large scale CPS.
- *CPS Engineering*, comprising methods and tools for the CPS life-cycle support, including the design, development and deployment.
- *CPS Ecosystems*, focusing the design and deployment of collaborative, autonomic, self-* and emergent CPS, as well as the integration of Humans in the Loop.
- *CPS Infrastructures*, related to interoperability services, and mitigation and migration strategies to support CPS infrastructures.
- *CPS Information Systems*, considering artificial intelligence and the transformation of data and information analytics to actionable knowledge.

Among the identified challenges, the use of bio-inspired mechanisms, and particularly self-organization, to support the dynamic evolution and adaptation to condition changes, is a promising issue to empower the deployment of such CPS solutions.

2.2 Self-Organization as the CPS Aggregation Factor

Evolution is the process of change, namely the development, formation or growth, over generations, leading to a new, more advanced or complex form [13]. Self-organization is one approach to achieve the dynamic system evolution, being defined as a process of evolution where the development of new, more advanced and complex structures takes place primarily through the system itself, and normally triggered by internal driving forces. Different types of self-organization can be observed in nature, e.g., stigmergy (e.g., used by ants), decrease of entropy (e.g., represented by the 2^{nd} law of thermodynamics) and autopoiesis (e.g., found in the cells reproduction), each one defining different driving forces to trigger the self-organization process.

Research in self-organization mechanisms are taking much attention, being applied to different domains, such as economics, sociology, computing, robotics and manufacturing. Particularly in manufacturing domain, self-organization allows the dynamic and on-the-fly evolution and re-configuration of the organizational control structure, supporting the agile reaction to unpredictable condition changes. For this purpose, the self-organization concept can be translated for the development of self-organized CPS, which allows to achieve truly reconfigurable systems that address the current industrial requirements. These self-organized CPS allow to reach the requirements of flexibility, robustness, adaptation, reconfigurability and agility, in a simple and intrinsic manner, with the system behaviour emerging from the interaction among individuals.

These systems are traditionally more difficult to design since complexity also comes from the non-linear interactions among individuals involving amplification and cooperation, and the sensitivity to initial conditions (i.e. the butterfly effect). In fact, the resulting behaviour is difficult to predict since the large number of non-linear interactions may cause a large number of possible non-deterministic ways the system can behave. For this purpose, the self-organization process should be properly controlled to guarantee that expected properties will actually emerge, and not expected and not desired properties will not emerge. Additionally, in such dynamic and self-organized systems, the existence of regulation mechanisms are crucial to control the system nervousness and to maintain the system in a stable state, avoiding the increase of entropy and chaos.

Self-organization was used in several works in manufacturing domain, namely in P2000+ [5], ADACOR [16], PROSA + ants [19], AirLiquide [17] and ADACOR^2 [2]. This last approach combines a behavioural self-organization perspective to ensure the smooth system's evolution (aligned with the Darwin's theory of evolution of the species) and the structural self-organization perspective to support the drastic evolution episodes (aligned with the punctuated equilibrium theory). However, the number of practical applications running in industrial environments is reduced or using weak self-organization implementations. A significant work should be performed to disseminate the potentialities of using self-organization in large and complex industrial CPS.

3 Modular Cyber-Physical Conveyor System

The case study considered in this work is related to a conveyor system composed by a set of modular Fischetechnik conveyors, each one having the same structure and offering the same functionalities, i.e to convey parts from the input to the output position. Physically, each individual conveyor is composed by a belt operated by a 24V DC motor. The detection of parts at the beginning and ending positions is achieved by means of independent photo-electric barrier sensors. Similarly to the motor, the sensors operate at the nominal voltage of 24V, making it industrially compatible, e.g., allowing their direct connection to a Programmable Logic Controller (PLC). In such system, the conveyor starts its motor when the part arrives to the output sensor of the previous conveyor, and stops its motor when the part arrives to the input sensor of the next conveyor.

The logical control of the aforementioned modular conveyor system can be implemented using several approaches. A traditional choice would be to use a centralized IEC61331-3 [9] control program running in a PLC to regulate the behaviour of the overall conveyor system, which lacks the conveyor system scalability and the re-configuration. In particular, the easy re-configuration of the conveyor system, e.g., add a new conveyor or switch the order of the conveyors, is complex and time-consuming. The use of the IEC61499 standard [10] presents interesting features, namely its distributed nature, but lacks the support for the intelligent and autonomous decision, which is crucial for the development of self-organization mechanisms. The challenge is to use MAS technology to achieve the self-organization and reconfiguration on-the-fly, i.e., without the need to stop, re-program and re-start the system components, which can not be easily achieved by traditional approaches.

In this work, the logical control of the conveyor module uses agents technology deployed into the Raspberry Pi boards. The Raspberry Pi control board is powered by a standard USB cable, and therefore uses a 5V power supply while its General Purpose Input Output (GPIO) ports are 3.3V compatible. This requires the need to develop an interface board, also commonly named as "shields", mainly responsible for:

– Supply the power to the conveyor belt and provide an isolated supply for the Raspberry Pi board.
– Connect physically the Raspberry Pi (processing part) and the conveyor belt (physical part).
– Indicate, by using LEDs, the system operation and the GPIO usage.

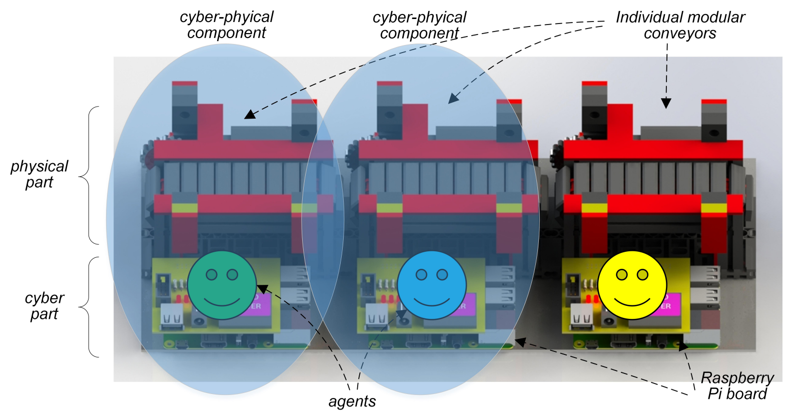

Fig. 1: Individual conveyor as a cyber-physical component.

In this way, the individual conveyor system is divided into two symbiotic parts, namely the logical part and the physical part. The logical part is provided by the processing capabilities offered by the agents running in the Raspberry Pi, constituting the *cyber* level, while the *physical* part is provided by the functions offered by the group formed by the conveyor, motor and sensors. By this, and as depicted in Fig. 1, the bundle constitutes a cyber-physical component, which combining different similar ones will constitute a cyber-physical system for the conveyor system.

4 Engineering the Modular and Self-Organized Conveyor System

This section presents the engineering of the modular conveyor system composed of several similar individual conveyors, namely describing the technical issues regarding the deployment of agents into the Raspberry Pi control boards.

4.1 Designing the Agents

The JADE (Java Agent DEvelopment Framework) [4] platform was used to implement the agents that will control the individual conveyors. One of the most important features of implementing the logical control by using the agent technology is the possibility of programming-once and deploying multiple times. Nevertheless, this interesting feature is only fully exploited when the agents are properly designed to be as generic as possible. In this way, the behaviour of each agent controlling an individual conveyor comprises a simple initiation process that involves the registration of the agent's skills in the yellow pages, the search of agents with similar skills and a general announcement of the conveyor presence in the system. This phase is finalized by launching the behaviours that are responsible to handle the message exchange, the internal logics and the self-organization process.

In the logical perspective, the behaviour of each agent relies on the control of the motor of its conveyor according to the input and output sensors and the synchronization with the behaviour of precedent and posterior conveyors, as represented by the Petri nets [18] model illustrated in Fig. 2. In fact, each conveyor must inform the adjacent conveyors in a two-fold manner. When the conveyed part is in the input sensor, the conveyor must inform the previous one (if any) that it has the part possession, allowing the previous conveyor to stop its motor. Similarly, when the part is in the output sensor, the conveyor must inform the successive conveyor (if any) that it should start its motor in order to properly receive the part.

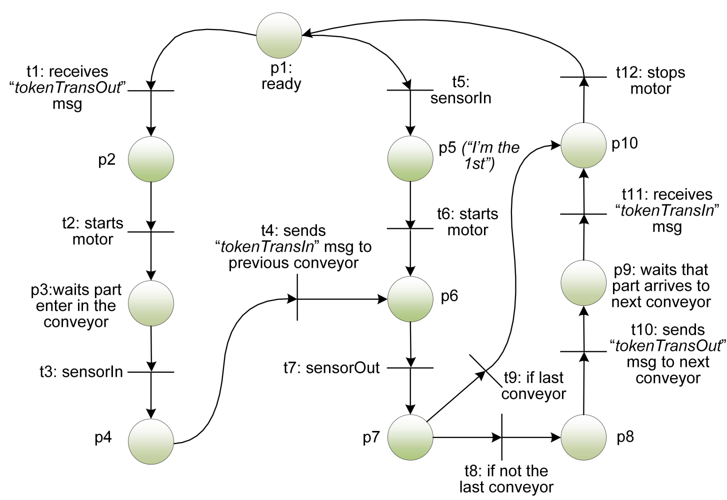

Fig. 2: Logic control for an individual conveyor agent

The management of this collaborative process among the conveyors is performed by the agents through the exchange of messages following the FIPA Agent Communication Language (ACL). The *tokenTransIn* message is used to inform the agent that the part is at the input sensor of the posterior conveyor and the *tokenTransOut* message is used to inform the agent that the part is at the output sensor of the previous conveyor. These messages contain, at their content, the reference to the *token*, i.e. the indication of the conveyor current position, which the receiving agents use to check if the token passage concerns to them, and if yes, stopping or starting the motors accordingly.

The agents exchange other types of messages during their cooperation processes, namely the *informIAmAlive*, *thereIsASwap*, *swapTokenFound* and *IAmLeaving*. These messages, governing several phases and situations in the conveyor behaviour, will be deeply explained during the description of the self-organization mechanisms.

4.2 Cold Start of the System

At the system set-up, none of the agents is aware of their positions in the system sequence, which requires the execution of a distributed procedure to determine the sequence of the individual conveyors. When a part is placed at a given conveyor, and the correspondent agent doesn't know its current position, the agent assumes the *token* value 1 (first conveyor in the system). Then, when the piece reaches the output sensor, a message is broadcasted to all conveyors in the system. At this stage, the conveyors that have the posterior token or don't know their positions will start their motors. Note that, the conveyors that already know their positions and are not the subsequent conveyors will ignore the message. The agent that receives the piece and don't know its current position in the sequence will update it accordingly to the passed token value. This procedure is repeated until the part reaches the last conveyor.

4.3 Deployment of Agents in Raspberry Pi

After the implementation phase of the agents, these must be deployed and executed in the Raspberry Pi boards. Due to the JADE inherent features, one main container must be initiated before the agents' execution to contain the agent platform. Despite the fact that one of the Raspberry Pi boards could be selected to host this main container, a cloud-based approach was selected to host the JADE agent-based platform. This means that no central governing mechanism is deployed into the cloud, i.e. the system is governed using a decentralized self-organization mechanism without providing any system topology, size or conveyor position into the agents. This decision allows to reach a more flexible approach permitting to remove any conveyor, releasing the user from the constant need of ensuring the main-container up-time.

The deployment of agents in the Raspberry Pi is straightforward, being only necessary to upload the agent package (i.e. the .jar file containing the agent instantiation) into a system directory. Despite this, several preparatory work in the Raspberry Pi board is advised to be performed, namely:

- The installation of a Java Run-time Environment (JRE).
- The installation of a Java API to allow the agent to access the GPIOs and/or to system information.
- The definition of the environment variables (to simplify the agent's execution from any point in the system).

The implementation uses the PI4J Java API (http://pi4j.com/) that bridges the Raspberry Pi kernel into Java compliant methods. Besides to allow the control of GPIOs, this API also allows the access to the serial communication to gather the system/network information and the creation of event listeners. An important remark: to have access to the hardware in the Raspberry Pi board, the agent needs to have *root* rights.

The access of the agent to the hardware is configured during its initialization behaviour where the required GPIOs are provisioned and configured properly, using the following excerpt of code:

```
myMotor = gpio.provisionDigitalOutputPin
                        (RaspiPin.GPIO_04, PinState.LOW);
outSensor = gpio.provisionDigitalInputPin
                        (RaspiPin.GPIO_05);
inSensor = gpio.provisionDigitalInputPin
                        (RaspiPin.GPIO_06);
```

Listeners are also added to govern the agent's actions triggered by the change of state in the input and output light-sensors. The following excerpt of code exemplifies the listener that was implemented for the *inSensor*.

```
inSensor.addListener(new GpioPinListenerDigital() {
  @Override
  public void handleGpioPinDigitalStateChangeEvent(
           GpioPinDigitalStateChangeEvent event) {
    // code to be executed by Raspberry Pi board when
    // the sensorIn signal changes its state goes here
  }
});
```

5 Mechanisms for Plugability and Self-Organization

An important issue during the operation of the modular cyber-physical conveyor is to ensure the presence of mechanisms that support the plugability and self-organization of the conveyor system on the fly, i.e. without the need to stop, re-program and re-start the individual components. For this purpose, this section details the implementation of self-organization mechanisms to support the system's operation in evolvable environments, namely adding a new individual conveyor, removing a broken individual conveyor or swapping two individual conveyors. The overall self-organization mechanism is then built by the composition of few simple rules (similarly to what happens in nature).

5.1 Plug-in and Plug-out of Individual Conveyors

The plug-in of a new conveyor can happen at different locations along the sequence, being required to design a simple mechanism for the automatic and decentralized detection of the conveyor position, as illustrated in Fig. 3.

Initially, the new conveyor agent registers its skills in the DF service and sends an *informIAmAlive* message to the other agents informing that it is ready to work. The other conveyor agents receiving this message are able to update the number of conveyors placed in the system. After this set-up phase, the new conveyor agent is waiting for the occurrence of one of two situations: i) the arrival of a part in the sensorIn sensor or, ii) the arrival of a *tokenTransOut* message. The occurrence of the first case, i.e. the detection of a true signal in the input sensor of the conveyor without receiving any *tokenTransOut* message, means that the new conveyor is placed in the beginning of the conveyor sequence, and after starting its motor, the conveyor agent will send a *tokenTransOut* message when the part reaches its output sensor. In the second case, i.e.

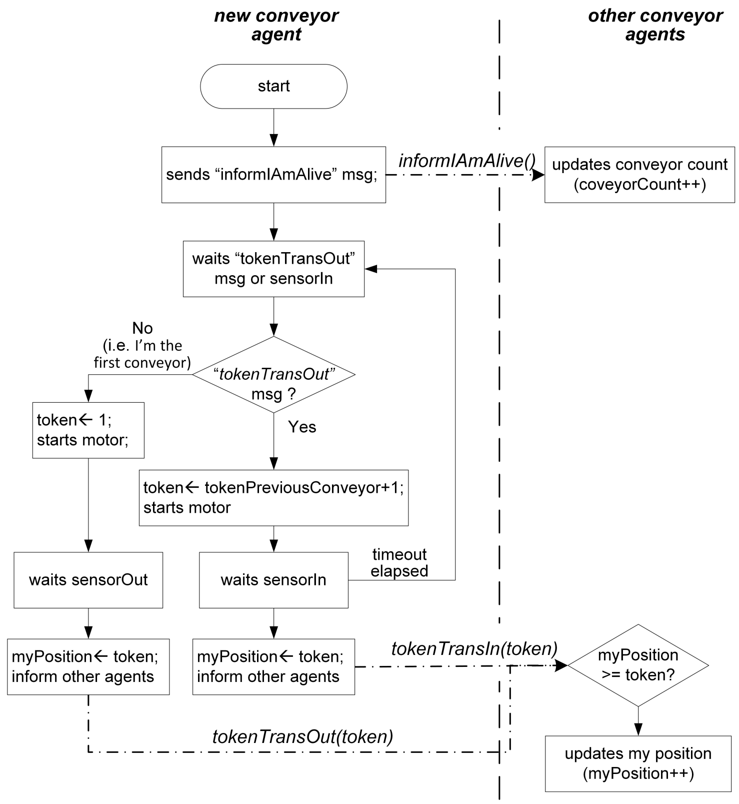

Fig. 3: Mechanism for the automatic plug-in of a new conveyor.

the arrival of the *tokenTransOut* message, the conveyor agent starts its motor and waits that the part arrives to its input sensor. If it occurs before a time-out, it means that the new conveyor is placed after the conveyor that has sent the *tokenTransOut* message. In this case, the new conveyor agent updates its location and sends a *tokenTransIn* message to the previous conveyor agent and a *tokenTransOut* message to the other conveyor agents when the part reaches its output sensor. In this case, only the conveyor agents that are located after the introduced new conveyor will update their positions.

If the new conveyor agent detects that it is located at the end of the sequence, it will stop its motor when the part reaches the output sensor and not when the *tokenTransIn* message from the posterior conveyor agent is arrived. In this case, the conveyor agents only update the individual conveyors counting without affecting the other agents.

Similarly, the removal of a conveyor is broadcasted by the leaving conveyor agent before its removal, as illustrated n Fig. 4.

The conveyor agents will decrease by one the number of conveyors placed in the system (i.e. decreasing the variable *conveyorCount*) and those placed after the removed conveyor will downward their locations by one position.

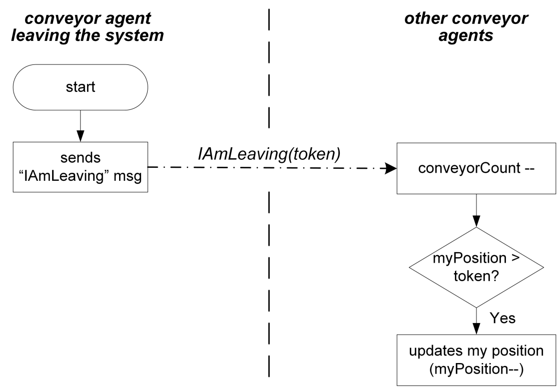

Fig. 4: Mechanism for the automatic plug-out of conveyors.

5.2 Change the Order Sequence of Conveyors

The swap mechanism requires a slightly different approach since the conveyor counting is kept and the new positions must be discovered. The self-organization mechanism to support the conveyor swap comprises several stages, as depicted in Fig. 5.

The first step is related to the detection phase. In normal operation, when a piece is at the output sensor, a message is broadcasted to all the agents and is processed by the succeeding conveyor agent and discarded by the rest. When the succeeding conveyor agent detects that the piece has not reached by its input sensor (by means of a timeout), it broadcasts the *thereIsASwap* message that warns for a possible order change (and containing the indication of the current position).

All the conveyor agents that have a token higher than the received position, will start their motors (naturally guaranteeing that they are in a valid situation, e.g., without have in possession a piece to convey). Afterwards, when a conveyor agent receives the piece at its input sensor, it will update its current system position and broadcasts a *swapTokenFound* message with its previous and new positions (updated with the position received during the *thereIsASwap* message). The other conveyor agents that receive this message only update their new positions.

This process is repeated every time a possible conveyor order has changed, governing, in a decentralized and self-organized manner, the system behaviour. The use of agents and self-organization principles allow, in a distributed and decentralized manner, to govern this conveyor system using a simple set of rules. The absence of a central control logic node also increases the system robustness and scalability by eliminating single-point of failure situations. Additionally, the use of agents allow to reach a truly "plug-and-produce" concept.

6 Conclusions

This paper describes a self-organized and modular CPS composed by several several individual cyber-physical conveyors. Each individual cyber-physical conveyor is com-

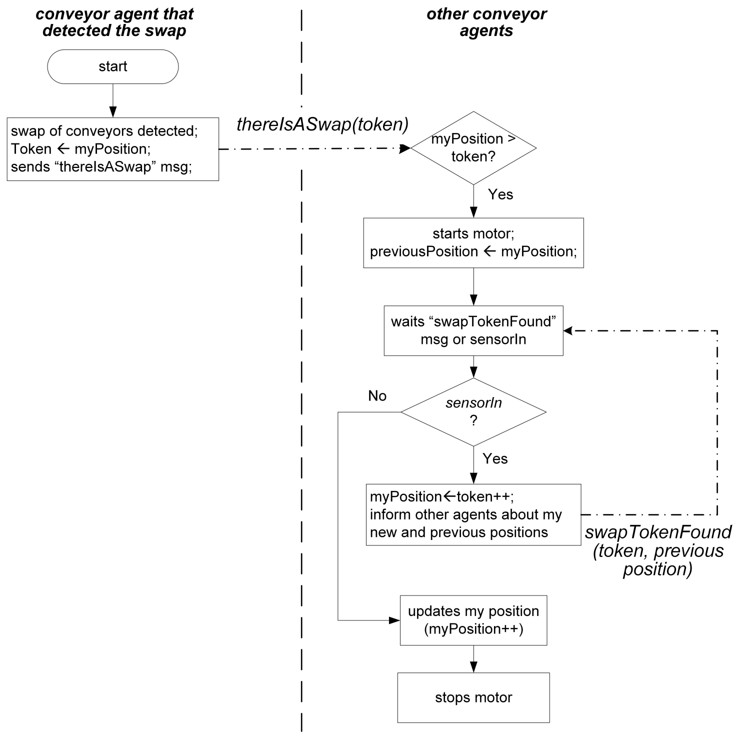

Fig. 5: Mechanism for the automatic swap of conveyors.

posed by a conveyor belt, constituting the physical part, and a logical part, implemented using agent technology and deployed in a Raspberry Pi board. Arranging together different individual cyber-physical conveyors allows to convey pieces from an initial position to the final position, by means of cooperation between all the conveyors, in a system of systems perspective

A simple self-organization mechanism was deployed, in a distributed manner and with MAS technology, to govern the system operation. This mechanism, comprising several simple rules, allowed the successful system operation, including the addition and/or removal of conveyors, or even the change of the sequence of conveyors on the fly, maintaining the system operability.

This CPS platform was used in the practical learning classes of the summer schools on intelligent agents in automation, held in Lisbon, Portugal in 2015 and posteriorly in Linköping, Sweden in 2016, providing the hands-on experience to the participants on deploying agents and developing simple self-organization mechanisms.

The described self-organized CPS is a simple example that shows the potentialities of applying CPS and self-organization concepts to industrial environments. The future work passes by the further refinement of the described self-organized mechanisms, namely refining the configuration process where diverts and convergences are needed.

References

1. ACATECH: Cyber-Physical Systems: Driving force for Innovation in Mobility, Health, Energy and Production. Tech. rep., ACATECH – German National Academy of Science and Engineering (Dec 2011)
2. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Dynamic Self-organization in Holonic Multi-agent Manufacturing Systems: The ADACOR Evolution. Computers in Industry 66, 99–111 (2002)
3. Bauer, H., Baur, C., Camplone, G., et. al.: Industry 4.0: How to Navigate Digitization of the Manufacturing Sector. Tech. rep., McKinsey Digital (2015)
4. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley (2007)
5. Bussmann, S., Schild, K.: Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS'00). pp. 87–94 (2000)
6. EFFRA: Factories of the Future, Multi-Annual Roadmap for the Contractual PPP under H2020. Tech. rep., European Commission (2013)
7. ElMaraghy, H.: Flexible and Reconfigurable Manufacturing Systems Paradigms. International Journal of Flexible Manufacturing Systems 17, 261–271 (2006)
8. Ferber, J.: Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)
9. IEC - International Electrotechnical Commission: IEC 61131: Programmable controllers – Part 3: Programming Languages (2012)
10. IEC - International Electrotechnical Commission: IEC 61499: Function blocks – Part 1-4 (2012)
11. Kagermann, H., Wahlster, W., Helbig, J.: Securing the Future of German Manufacturing Industry: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0. Tech. rep., ACATECH – German National Academy of Science and Engineering (2013)
12. Lee, E.A.: Cyber Physical Systems: Design Challenges. In: Proc. of the 11^{th} IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'08). p. 363–369 (2008)
13. Leitão, P.: Holonic Rationale and Bio-inspiration on Design of Complex Emergent and Evolvable Systems. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) Transactions on Large Scale Data and Knowledge Centered Systems, Lecture Notes in Computer Science, vol. 5740, pp. 243–266. Springer (2009)
14. Leitão, P.: Agent-based Distributed Manufacturing Control: A State-of-the-art Survey. Engineering Applications of Artificial Intelligence 22(7), 979–991 (2009)
15. Leitão, P., Karnouskos, S., Colombo, A.W.: Industrial Automation based on Cyber-Physical Systems Technologies: Prototype Implementations and Challenges. Computers in Industry 81, 11–25 (Sep 2016)
16. Leitão, P., Restivo, F.: ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control. Computers in Industry 57(2), 121–130 (2009)
17. Miller, P.: The Genius of Swarms. National Geographic (July 2007)
18. Murata, T.: Petri Nets: Properties, Analysis and Applications. IEEE 77(4), 541–580 (1989)
19. Valckenaers, P., Hadeli, Kollingbaum, M., Brussel, H., Bochmann, O.: Stigmergy in Holonic Manufacturing Systems. Journal of Integrated Computer-Aided Engineering 9(3), 281–289 (2002)