A Novel Implementation Approach for Resource Holons in Reconfigurable Product Manufacturing Cell

Ahmed R. Sadik[1,2] and Bodo Urban[1,2]
[1]University of Rostock, Universitätsplatz 1, 18055, Rostock, Germany
[2]Fraunhofer Institute for Computer Graphic Research IGD, Joachim-Jungius-Str. 11, 18059, Rostock, Germany

Keywords: Holonic Control Architecture, Reconfigurable Manufacturing System, Human-Robot Cooperation Planning, Autonoumos Multi-Agent System, IEC 61499 Model.

Abstract: Holonic Control Architecture is a successful solution model for reconfigurable manufacturing problems. Two well-known different technologies have been used separately to implement the holonic control model. The first technology is IEC 61499 standard, and the second is autonomous reactive agent. Both of the previous mentioned technologies have its own pros and cons. Therefore this research is merging the two technologies together in one solution body, to magnifying their pros and reduce their cons. Ultimately; it provides a novel implementation model for the manufacturing holons, to be followed in similar reconfigurable manufacturing problems. A human worker in cooperation with a safe industrial robot, has been selected as a case study of a reconfigurable manufacturing problem. The proposed holonic control solution has been applied to the case study, to evaluate the ability of the solution to satisfy the requirements of the case study. The results show the ability of the proposed control solution to provide a flexible physical and logical interaction framework, which can be scaled over more workers in cooperation with more industrial robots.

1 INTRODUCTION

Reconfigurable Manufacturing System (RMS) is a system where production components and functions can be modified, rearranged and/or interchanged in a timely and cost-effective manner to quickly respond to production requirements (Koren, 1999). The concept of RMS has been formed in response to the fast continuous changes in the market requirements. The goal of the RMS is to reduce the lead time when the production switches from one product to another, moreover to reliably handle the fluctuation in the production volume (Kruger, 2015). A Reconfigurable Manufacturing Cell (RMC) is the elementary unit of an RMS. An RMC should be able to produce different customized products, this can be achieved if the RMC control system understands the product task plan (i.e. recipe), and the capabilities of the work resources (i.e. machines, robots, workers). Then it matches the product task plan to the capabilities of the work resources.

A Holonic Control Architecture (HCA) is an agile solution for RMS problems (Chirn, 2000). The solution provides a distributed control model which defines the hierarchy, structure, and functions of all the elements in a manufacturing system. HCA is based on the autonomy concept. Autonomy is the ability of the system to act without a direct intervention from humans, as it should have control over its own actions and internal state (Anumba, 2005). Many different autonomous technologies can be used to implement the HCA model. Most of the researchers who implement the HCA model, either use autonomous agent specifications (Wang, 2005) or IEC 61499 standard (Vlad, 2010). During this research both the technologies will be highlighted. Then they will be combined together to get the best possible solution, to implement an HCA model to control an RMC. A safe industrial robot in flexible cooperation with a human worker is a novel case study of an RMC, the proposed HCA solution will be tested on this case study to show the solution viability.

This paper is structured as follows. Section 2 explains the main concepts and technologies that will be utilized during this research, such as HCA, autonomous agent technology, and IEC 61499 standard. Section 3 is a review for the most related work. Thus it is easier for the reader to understand the problem formulation in section 4. Section 5

Sadik, A. and Urban, B.
A Novel Implementation Approach for Resource Holons in Reconfigurable Product Manufacturing Cell.
DOI: 10.5220/0005956801300139
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 1*, pages 130-139
ISBN: 978-989-758-198-4

introduces an RMC case study and the proposed solution structure, which is tested and evaluated in section 6. Finally section 7 wraps up the work summary with the conclusion and the future research.

2 PRELIMINARY CONCEPTS

2.1 Holonic Control Architecture

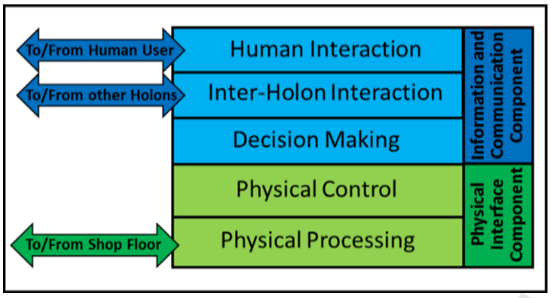

Figure 1: Holon General Architecture.

In the late of sixties, the term holon has been introduced for the first time by philosopher Koestler (Koestler, 1967). Koestler developed the term as a basic unit in his explanation for the evolution of biological and social structures. Based on his observations that organisms (e.g., biological cells) are autonomous self-reliance units, which have a certain degree of independent control of their actions, yet they still subject to higher level control instructions. His conclusion is that any organism is a whole "holos" and a part "on" in the same time, which derived the term holon (Giret, 2008). The concept of holon has been adopted in the early of nineties by the Intelligent Manufacturing Systems (IMS) consortium, to define a new paradigm for the factory of the future. IMS defined the holon as an autonomous cooperative building block of the manufacturing system, that can be used to transform, transporting, store and/or validate the information and the physical objects (Radu and Frank, 2006).

The generic structure of the holon has been constructed as shown in Figure 1 (Bussmann, 1998). A holon is mainly composed of two components. First the physical interface component, this is the component which physically connects the holon to the automation devices input/output (I/O) on the factory shop floor. The holon physical component is responsible for controlling the data transfer from the automation devices to the holon and vice-versa. Furthermore it translates these data into useful information which can be processed by the holon. The second holon component is the information and communication component, this component contains the holon kernel. As it is responsible for communicating with other holons in the HCA, and interacting with the humans in the manufacturing system. All the information gathered from the other holons, the humans, and the automation devices are processed by the holon decision making algorithm, to produce a proper action based on the function of this holon in the HCA.

The functions and the responsibilities of each holon, depend on the category of this holon. Product-Resource-Order Staff Architecture (PROSA) (Van Brussel, 2003), and ADAptive holonic COntrol aRchitecture (ADCOR) (Leitao, 2006) are the most popular HCA models. Regardless the difference between the two models, they are addressing four holon catagories which are: resource holon, product holon, task holon, and supervisory holon (Su, 2007). However our concern in this research will focus on the resource holons. The resource holon is a physical entity within the manufacturing system, it can represent a robot, conveyor, machine, etc.

2.2 IEC 61499 Function Block Standard

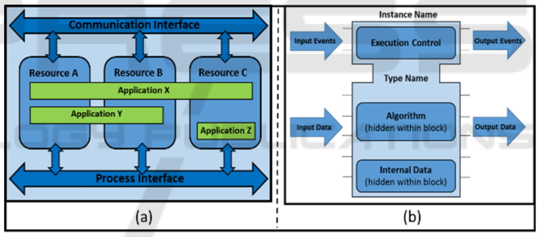

Figure 2: (a) IEC 61499 Distributability – (b) IEC 61499 Function Block.

IEC 61499 standard is an implementation model for distributed control systems on embedded devices, which extends the capabilities of IEC 1131-3 standard. It defines a reference model for development, reusing and deployment of Function Blocks (FBs) in distributed embedded industrial control (Fischer, 2010). Figure 2-a illustrates the distributability of IEC 61499. The automation devices share the control of different applications simultaneously. IEC 61499 power of distributability originates from using the FB as a main building unit. Figure 2-b shows the structure of an FB, The FB consists of three fields: events I/O, data I/O, and an internal algorithm. An FB does not apply scanning cycle technique. Therefore different algorithms can run simultaneously on the same FB. An algorithm will be triggered on arrival of an input event, the algorithm is processing the input data to produce output data and output events (Pang, 2014).

FB technique does not only empower the customizability and the distributability concepts. But also it supports other concepts such as: reusability, modularity, extensibility, and diagnosability. Reusability points to the capability of the software components to be reused over different manufacturing demands and configurations. While modularity is the degree of those software components to be separated and recombined (Sturgeon, 2002). Extensibility is another synonymous of “plug and play” concept, it can express the ability of the system to be extended to intra/inter enterprise level by including or integrating with more manufacturing functions (Su, 2007). Finally diagnosibility is the ability to automatically read the current state of a system and controls, so as to detect and diagnose the operational defects, and subsequently correct them quickly (Koren, 2005).

Many software tools are using IEC 61499 as a reference control architecture. Examples of those tools are FBDK, ISaGRAF, 4DIAC, nxtStudio. Those tools differ in minor variations, such as the programming language they build in. This research is using Function Block Development Kit (FBDK) as part of the implementation and testing phase. FBDK is not only an IEC 61499 development tool, but also it can be used to emulate the manufacturing system before the deployment phase. Manufacturing system emulation is a real time simulation, which refers to the ability of the logical model to identically imitate the time responses and behaviours of the real physical manufacturing system, to gain insight into it, before the actual implementation (Peters, 1996).

2.3 Autonomous Agent Technology

A software agent is a computer system situated in a specific environment; that is capable of autonomous actions in this environment in order to meet its design objective (Jennings, 1998). An agent is responsive, proactive and social. Responsive means the agent can perceive its environment and respond in a timely fashion to the changes occurring in it. Proactive means the agent is able to exhibit opportunistic, goal directed behaviour and take initiative. Social means the agent can interact with other artificial agents or humans within its environment in order to solve a problem. A Multi-Agent System (MAS) is a collective system composed of a group of artificial agents, teaming together in a flexible distributed topology, to solve a problem beyond the capabilities of a single agent (Shen, 2006). An agent can be designed based on a reactive or cognitive model. Reactive agents have knowledge compiled from the actions to be carried out. They do not need to construct a mental representation of their environment, since it is merely sufficient for them to react to the situations (Wooldridge, 1995). On the other hand cognitive agents have reasoning capacity based on their environment representation. They are capable of memorizing situations, and analysing them. To foresee possible reactions to their actions, thus they can plan their behaviours (Franklin, 1998).

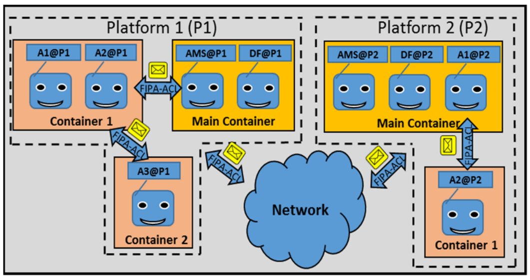

Figure 3: Java Agent Development (JADE) Framework.

JAVA Agent Development (JADE) is a distributed MAS middleware framework (Bellifemine, 2007). JADE applies reactive agent architecture which complies with the Foundation for Intelligent Physical Agent (FIPA) specifications, and provides a graphical interface to deploy and debug a MAS (Italia, n.d.). JADE agents use FIPA-Agent Communication Language (FIPA-ACL) to exchange messages either inside its own platform or with another platform in a distributed MAS. Figure 3 shows a scheme example of JADE (Poslad, 2007). Each JADE instance is an independent thread contains a set of Containers. A Container is a group of JADE agents run under the same JADE runtime instance. Every platform must contain a Main Container. A Main Container contains two necessary agents which are: an Agent Management System (AMS) and a Directory Facilitator (DF). AMS provides a unique ID for every agent under its platform, to be used as an agent communication address. While the DF announces the services every agent can offer under its plateform, in order to facilitate agent service exchange, so that each agent can obtain its specific goal (Caire, 2009). A JADE agent can implement one or a group of FIPA standard behaviours. A behaviour is a set of action/reaction routines triggered by the agent due to its perception for its environment (Teahan, 2010).

3 RELATED WORK REVIEW

The research in (Vlad, 2010) is proposing an IEC 61499 solution for modeling and implementing holons dynamic interaction. The paper used FBDK visualization capabilities to emulate the holons interaction dynamics. The research also is providing an extensive guidance for designing both the physical and communication components of a resource holon, based on a comprehensive explanation of the responsibilities of each holon category within the manufacturing system. The holon communication component has been implemented using FBDK publish/subscribe technique, which is very static and primitive communication approach. Therefore the paper was very successful to emulate the holons physical dynamics, however it failed to show any communication interaction. Also the paper did not apply its described theoretical solution to any practical case study.

(Kruger, 2013) is evaluating two different solution approaches. One uses IEC 61499 and the other uses agent technology. To implement a holonic control for modular feeder subsystem of an experimental reconfigurable assembly system (RAS). The research does not combine the two technologies together, but it compares them. Then it evaluates them based on the qualitative and quantitative performance of four different hardware configurations of the RAS. The research results state that agent technology requires less effort and time than IEC 61499 to implement three configurations of the RAS. The IEC 61499 has inherent simplicity because of its visualization capabilities, however the lacks in IEC 61499 inter-platform communication flexibility restricted the hardware reconfigurability.

An Intelligent control of an Airport Baggage Handling System (BHS) has been deeply studied in (Black, 2008). It provides an extended customized MAS solution, by modeling every component in the BHS as an IEC 61499 FB. Basically it is dividing the whole conveyor transportation system into many segments. Every conveyor segment is presented as a customized FB. Moreover it models every bag over the BHS as another customized autonomous FB. All the instances of the conveyor segment FB and the bag FB are communicating together using IEC 61499 standard communication FBs, which made the final model so crowded and complex. Two important points are distinguishing this research. First the reusability of one customized FB to handle all the instances of the conveyor segments or bags. Second is the viability of the solution to be practically applied, as finally it has been implemented on IEC 61499 FB compliant controller produced by TCS-NZ.

4 PROBLEM STATEMENT

The main focus of this paper is to design the resource holons in an RMS. The resource holons function is to autonomously adapt to the product changes, therefore the manufacturing cell is able to continuously produce the product without stopping. The resource holon design must meet finite criteria to achieve RMS concept. Those criteria are: distributability, customizability, reusability, modularity, extensibility, scalability and diagnosability. Also the holon design should achieve the interoperability concept, this means that the resource holons can communicate with each other regardless the running operating system (OS). Finally they should be capable of handling the shop floor I/Os and anticipating them to useful information. Most of other researches tried to design HCA either using IEC 61499 standard or intelligent agent based system. However each of those solutions has its own advantages and drawbacks (Vlad, 2010; Kruger, 2013; Black, 2008).

IEC 61499 offers a standard component-based framework, which promotes the encapsulation concept. Therefore it can be easily scaled, moduled, and integrated. Furthermore IEC 61499 is a modeling language which allows the designer to offline emulate his solution, before transferring it to the hardware. This gives The IEC 61499 model very high diagnosability characteristic comparing to other distributed control frameworks. However IEC 61499 does not provide an intelligent communication services, therefore it lacks of a high degree of adaptability, and this leads to increase the complexity of the overall solution model. Moreover IEC 61499 model can be customized in offline mode only to solve one version of a pre-known problem.

On the other hand, agent based system is far superior to IEC 61499 in information communication, as it characterized by a high level of flexibility and low degree of complexity. Agents are using standard communication languages to enable them to cooperate, collaborate, or negotiate. Furthermore an agent based system is using service oriented architecture (SOA) to announce the services every agent can offer within its platform. Therefore adaptability can be easily achieved. Nevertheless an agent based system can be very sophisticated when it deals with external physical control events.

In a plain English, the drawbacks of both of the control models can be compensated by merging them in a win-win paradigm. The missing gap in IEC 61499 model concerns adaptability, will be imported from the agent model. Yet the agent system will be privileged by the superiority of IEC 61499 model to deal with sophisticated I/O events and data.

5 CASE STUDY AND SOLUTION STRUCTURE

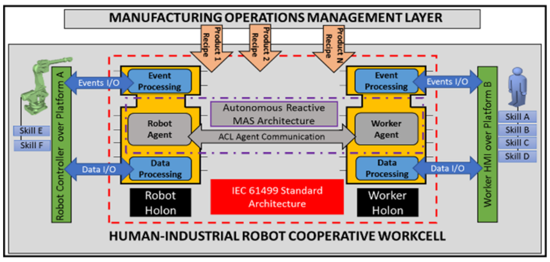

Figure 4: Holonic Human and Industrial Robot Cooperative Production Cell.

Human & Industrial Robot (HIR) cooperative workcell is a novel trend in industry, which integrates the human worker along with the industrial robot in the same working space, therefore they can perform together a cooperative industrial process (Nilsson, 2005). The idea behind this integration is to combine the advantages of both the human and the robot in the same production cell. Workers add the flexibility to the production cell, as they can easily adapt to many unexpected situations. Yet industrial robots are reliable in terms of speed, payload, and accuracy. Furthermore many industrial robot manufacturers are supporting this novel trend by producing safe industrial robots, such as KUKA lightweight, Rethink Baxter, YuMi ABB dual arm, and Universal Robots (Lasota, 2014). An HIR cooperative workcell is a direct application of the RMS. Thus the research will address it as a case study to apply an HCA solution.

Figure 4 shows the proposed HCA solution on a single robot – single worker cooperative workcell. The holon physical interface component implemented using IEC 61499 FB, while the communication component implemented as an autonomous reactive agent. Two different resource holons have been implemented, a robot holon and a worker holon. A customized IEC 61499 FB handles all I/O events and data to/from the robot controller, another customized FB handles the worker user interface I/Os. Each FB is embedding a reactive autonomous agent, the robot agent is communicating with the worker agent to form a simple MAS. The two holons are using FIPA-ACL standard protocol for sending/receiving their interaction messages. Each holon is dynamically acquiring its resource skills. Concurrently the robot holon is acquiring the product recipe. Thus the robot agent is able to negotiate with the worker agent by following a searching algorithm, to distribute the product recipe tasks between them. The product recipe within the context of this case study refers to, the order of the tasks to produce a certain product. While a skill refers to the capability of the work resource to perform a certain task.

For demonstration it has been assumed that the resources skills are unique, in other words the robot and worker do not share any similar skill. For the same reason, we selected a simple case scenario for one worker in cooperation with one industrial robot, only to show the applicability of the HCA solution. The solution approach can be extended to more than one worker or robot. The same issue with the searching algorithm, it can be more sophisticated based on the case study size and requirements. However the searching algorithm is beyond the paper goals, we just want to show the ability of the HCA solution to implement an algorithm, to provide some of the research goals such as adaptability.

5.1 Holon Physical Component Design

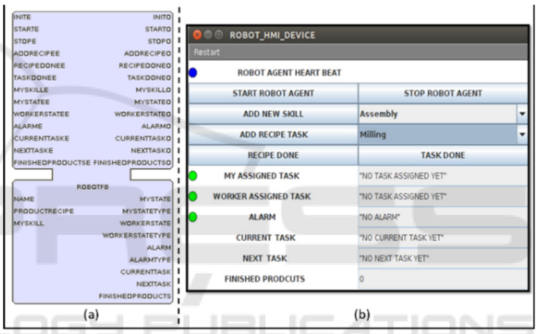

Figure 5: (a) Robot FB – (b) Robot I/O Interface.

FBDK is a pioneer software tool which applies IEC 61499. It is used under the umbrella of this research, to implement the physical component for the holon. Figure 5-a&b shows the design of a robot FB and a running interface of it respectively. INITE event is meant to initialize the robot FB input data. STARTE starts an agent in JADE, which holds the NAME input of the FB, while STOPE vanishes this agent. ADDRECIPEE event captures a PRODUCTRECIPE input. One product is composed from many tasks, which can vary from one product to another. RECIPE DONE indicates that a complete product recipe has been defined. TASKDONEE informs the FB that an assigned task has been accomplished. MYSKILLE is an event associated with MYSKILL input to capture the robot skills, considering an industrial robot can perform more than one task.

MYSTATEE is associated with MYSTATE and MYSTATETYPE outputs, to constantly display robot current state (e.g. Busy/Free and robot assigned task). Similarly WORKERSTATEE is associated with

WORKERSTATE and WORKERSTATETYPE outputs, to constantly display worker current state. ALARME is linked with ALARM and ALARMTYPE outputs, to occasionally display any alarm that could occur during the cooperation (e.g. required skill is not found within the production cell). CURRENTTASKE and NEXTTASKE are associated with CURRENTTASK and NEXTTASK outputs respectively, to display the current task in execution and the next task needed to be executed. Finally FINISHEDPRODUCTSE and FINISHEDPRODUCTS output are linked to display the number of produced products.

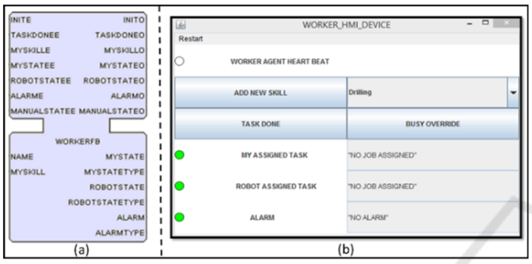

Figure 6: (a) Worker FB – (b) Worker I/O Interface.

Worker FB design and interface are shown in Figure 6. The difference between the worker and the robot FB, is that the robot FB defines the product and the robot I/O parameters together. However the worker FB concerns only the worker I/Os. One extra event has been added to the worker FB which is MANUALSTATEE, to be triggered manually by the worker if he is busy. Another difference is that the worker FB has been designed and deployed over Windows OS, while the robot FB has developed and deployed over Linux OS, to test/verify the interoperability of the solution on different OSs.

5.2 Holons Communication Algorithm

A communication algorithm is needed in order to design the holon communication component. Based on this communication algorithm, the agent behaviours can be programmed. Figure 7 illustrates an algorithm implemented with JADE to manage the cooperation between an industrial robot and a human worker, to produce variable recipe product. The algorithm starts automatically after defining the product recipe and both of the robot and the worker skills. In the very beginning it compares the product first task to the robot skills. If the task matches one of the robot skills, it assigns this task to the robot, changes the robot status to be busy, and informs the worker that this task has been assigned to the robot.

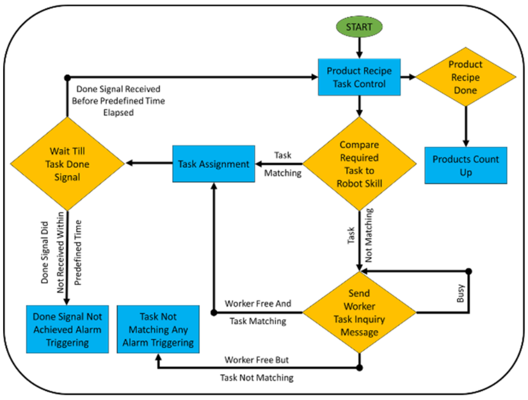

Figure 7: Interaction Algorithm between a Worker and an Industrial Robot.

In case the robot does not have the required skills to perform a product recipe task, the algorithm will send to the worker an inquiry holding the name of the required skill. If the worker is busy, the message will be sent periodically until he becomes available. If the worker is available and the proposed task matches one of his skills, the task will be assigned to him and his status will turn to be busy. All the information about the task and worker status will be sent to the robot FB. The algorithm will wait for a done signal, to proceed in the recipe order. If the worker is free but he does not have the inquired skill, an alarm will be raised to announce the skill shortage.

A more sophisticated cooperation algorithm has been intentionally avoided, otherwise it will be so hard for the reader to see the cooperation test results. Particularly, we offer a very generic solution, which can implement different algorithm variations.

6 SOLUTION TESTING

To verify the applicability of the proposed HCA, a simple cooperation scenario will be emulated, to show the ability of the solution to achieve the HCA definition. In this testing scenario we will order a product using the robot FBDK interface. The product recipe composes of three tasks, which are in order Drilling, Milling and Assembly. We will add Drilling as a robot skill and Milling as a worker skill. Intentionally we will not add Assembly skill either for the robot or the worker, to test the system behaviour in case of a skill shortage. Figure 8 and 9 show the first task of the product recipe (i.e. Drilling) assignment process.

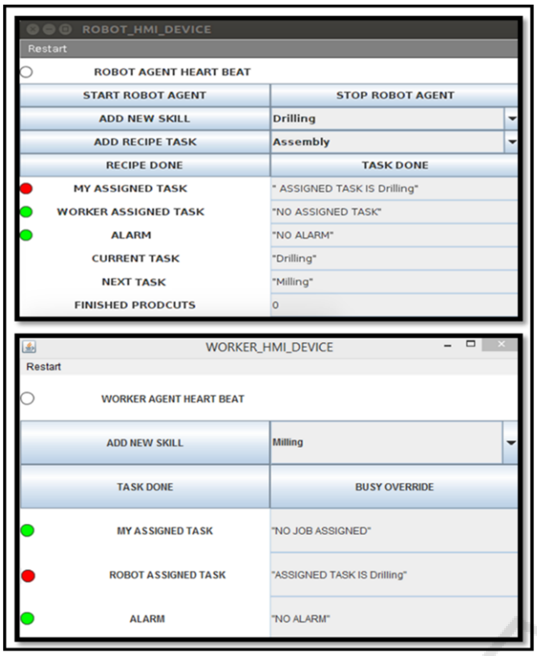

Figure 8: FBDK Interface for Drilling Task Assignment.

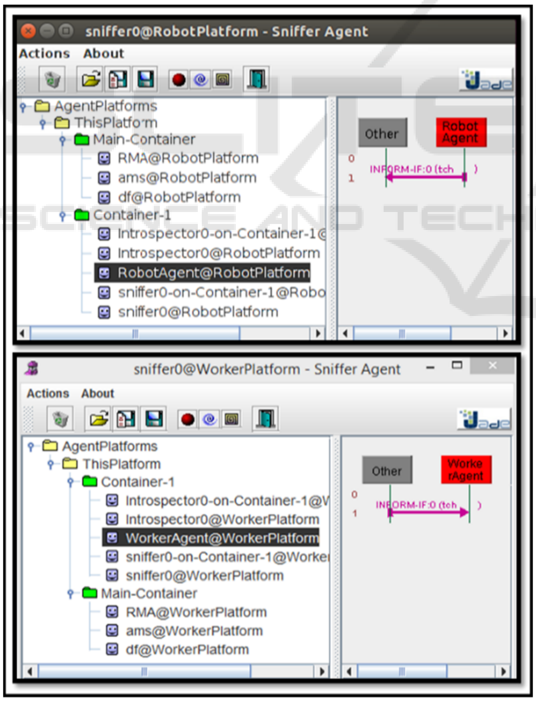

Figure 9: JADE Interaction for Drilling Task Assignment.

As been mentioned previously in the cooperative algorithm, the first step in the cooperation algorithm is to compare the robot skills with the product recipe current task. When a Drilling task matches the robot skills as shown in Figure 8, the MAS assigns the Drilling task to the robot. Also it changes its status to be busy, and sends an INFORM-IF message to the worker, to inform about a task assignment to the robot. Figure 9 shows the sending/receiving process between the robot and the worker agents, over two different OSs. Messages sending/receiving process is accomplished as part of the agent behaviours.

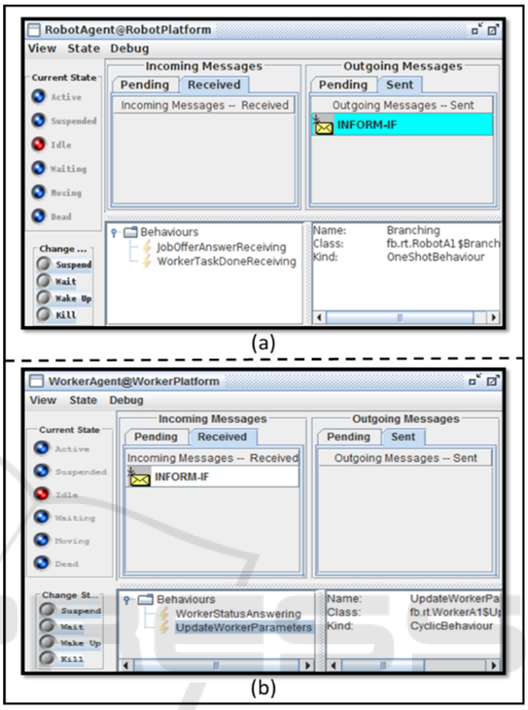

Figure 10: (a) ACL Message Sent by the Robot Agent – (b) ACL Message Received by the Worker Agent.

Figure 10-a shows three robot agent behaviours: Branching, JobOfferAnswerReceiving and WorkerTaskDoneReceiving. Branching behaviour is a one shot behaviour which means it is executed once when called (Ricci, 2011). Branching behaviour checks if the task matches the robot. If so, it sends an INFORM-IF message, which is received by the worker agent using UpdateWorkerParameters cyclic behaviour showing in Figure 10-b.

A cyclic behaviour is running all the time periodically. If the task did not match the robot skill, branching algorithm will offer the task to the worker agent using QUERY-IF message. The worker agent will receive QUERY-IF message then answers it with INFORM-REF using WorkerStatusAnswering cyclic behaviour. INFORM-REF will be received using JobOfferAnswerReceiving behaviour on the robot side. An INFORM message will be sent from the worker agent to the robot agent, to inform that a task has been done, WorkerTaskDoneReceiving cyclic behaviour is responsible for receiving this message.

Figure 11: (a) Robot One Product Cycle JADE Interaction – (b) Worker One Product Cycle JADE Interaction.

Figure 11-a&b show an overall worker-robot FIPA interaction, to produce one product described in the testing scenario. The rest of the task assignments mentioned in the test scenario are shown in Figure 12 and 13. The same previously described mechanism of message exchanging via agent behaviours is applied. In Figure 12-a and 13-a; the Milling task matches the worker skills and did not match the robot skills. Therefore it is assigned to the worker. In Figure 12-b and 13-b an Assembly task neither matches the robot nor the worker skills. Thus an alarm has been triggered in both the worker and robot FBDK interfaces. Furthermore the MAS keeps looking for an Assembly skill, till it is available in the production cell. Figure 12-c and 13-c show the reaction of the system when adding an Assembly to the worker skills, the MAS clears the alarm and assigns the Assembly task to the worker.

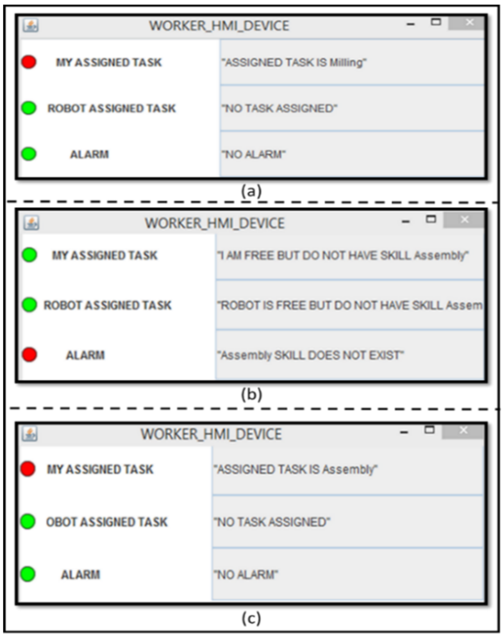

Figure 12: (a) Worker Milling Task Assignment Interface – (b) Worker Assembly Task Alarm Interface – (c) Worker Assembly Task Assignment Interface.

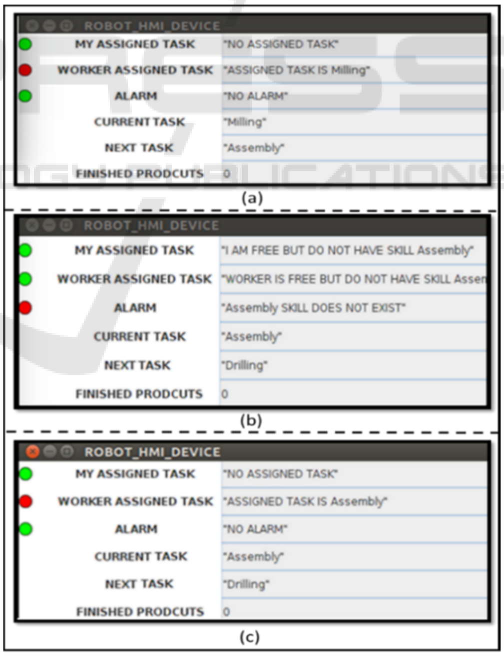

Figure 13: (a) Robot Milling Task Assignment Interface – (b) Robot Assembly Task Alarm Interface – (c) Robot Assembly Task Assignment Interface.

7 SUMMARY, CONCLUSION AND FUTURE WORK

The goal of this research is to design an RMC control system, which can adapt to a variable product recipe. A new product can be introduced at any time to the RMC, thus the work resources can rearrange their cooperation to produce the product without stopping the production process. An industrial robot in cooperation with a human worker has been proposed as an RMC case study. An HCA with two resource holons has been chosen as a proper control solution to be applied to the case study.

The research solution has combined two well-known control architectures that have been applied individually in previous researches as a design model for the HCA. The two architectures are IEC 61499 FB specifications and autonomous reactive agent architecture. IEC 61499 FB has been used to build a customized physical component for the holon. While reactive agent architecture has been used as an intelligent communication component. Merging the two architectures together in one solution, maximized their pros and minimized their cons. Therefore the solution successfully achieved the characteristics, which are highly required to deal with the mentioned RMC problem.

Modularity, reusability, customizability, extensibility, and diagnosability are the first set of characteristics obtained by this solution. Modularity has been achieved as the robot and the worker holons have two independent designs, therefore they can physically and logically separated. Furthermore the same holons can be reused and extended as an inherited property from IEC 61499 FB and agent architectures. Also the holons can be easily customized to fit more workers or robots into the interaction, which guarantee the extensibility of the solution. Diagnosability has been used during the testing phase, as a part of emulating the control system. Different combinations of the product recipe and the worker/robot skills inputs can be tested. Simultaneously the communication between the agents can be tracked within every different test.

Platform-independency is one of the most important results obtained by this solution. As it is shown in the testing phase, an MAS is formed between a robot holon running on Linux OS and a worker holon running on Windows OS. Thus interoperability, scalability, distributability, and integrity are no longer a problem during the final deployment phase. Finally the solution successfully achieved the adaptability concept during the testing scenario emulation. The MAS was able to adapt the existing work resources skills to a variable product recipe definition, and deal with some unexpected ambiguous situations such as skill shortage in the workcell.

Even the tested case study and its applied algorithm were simple, the required solution characteristics were completely fulfilled. Therefore the solution can be a good base model to follow, in solving more sophisticated interaction scenarios, with more cooperative work resources. In the future work more complicated testing case scenarios and algorithms can be tested and verified.

REFERENCES

Anumba, C. J., Ugwu, O., Ren, Z., 2005, *Agents and Multi-agent Systems In Construction*, Taylor & Francis Group.

Bellifemine, F., Caire, G., Greenwood, D., 2007, *Developing multi-agent systems with JADE.* Vol 7 ed. John Wiley & Sons.

Black, G., Vyatkin, V., 2008, 'Intelligent Component-Based Automation of Baggage Handling Systems with IEC 61499', *IEEE Transactions on Automation Science and Engineering,* Vol 7, pp. 699–709.

Bussmann, S., 1998, An agent-oriented architecture for holonic manufacturing control, *Proceedings of the 1st international workshop on Intelligent Manufacturing Systems,* pp. 1-12.

Caire, I., 2009, *JADE Tutorial: JADE Programming for Beginners*, TILAB & West Sussex.

Chirn, Jin-Lung, C.McFarlane, Duncan, 2000, A holonic component-based approach to reconfigurable manufacturing control architecture, *11th International Workshop on Database and Expert Systems Applications (DEXA'00),* pp. 219-223.

Fischer, J., Boucher, T.O., 2010, Workbook for Designing Distributed Control Applications using Rockwell Automation's HOLOBLOC Prototyping Software, Working Paper No. 05-017.

Franklin, S., Kelemen, A., McCauley, Lee, 1998, IDA: a cognitive agent architecture, *IEEE Systems, Man, and Cybernetics,* Vol 3, pp. 2646-2651.

Giret, A., Botti, V., 2008. Holonic Manufacturing Systems, *ANEMONA - A Multi-agent Methodology for Holonic Manufacturing Systems, Springer Series in Advanced Manufacturing*, pp. 7-20.

Italia, T., n.d., *jade.* [Online] Available at: http://jade.tilab.com/.

Jennings, N.R., Wooldridge, M.J., 1998, Applications of intelligent agents, *Agent Technology: Foundations, Applications, and Markets,* Springer, pp. 3–28.

Koestler, A., 1967, *The Ghost in the Machine,* Hutchinson.

Koren, Y., 2005, Reconfigurable Manufacturing and Beyond, *Keynote Paper - CIRP 3rd International conference on Reconfigurable Manufacturing.*

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Brussel H., 1999, Reconfigurable Manufacturing Systems, *CIRP Annals,* Vol 48, pp. 527-540.

Kruger, K., Basson, A., 2015, Implementation of an Erlang-Based Resource Holon for a Holonic Manufacturing Cell, *Studies in Computational Intelligence*, Vol 594, pp. 49-58.

Kruger, K., Basson, A., 2013, Multi-Agent Systems vs IEC 61499 for Holonic Resource Control in Reconfigurable Systems, *Forty Sixth CIRP Conference on Manufacturing Systems,* Vol 7, pp. 503–508.

Lasota A., Rossano F., Shah A., 2014, Safe Close-Proximity Human-Robot Interaction with Standard Industrial Robots, *IEEE International Conference on Automation Science and Engineering (CASE).*

Leitao, P., Restivo, F., 2006, ADACOR: A holonic architecture for agile and adaptive manufacturing control, C*omputers in Industry,* Vol 57, pp. 121-130.

Nilsson, K., Johansson, R., Robertsson, A., Bischoff, R., Brogårdh T., Hägele M., 2005, Productive robots and the SMErobot project, *Third Swedish Workshop on Autonomous Robotics*.

Pang, C., Patil, S., Yang, Chen-Wei, Vyatkin, V., Shalyto, A., 2014, a Portability Study of IEC 61499: Semantics and Tools, *12th IEEE International Conference on Industrial Informatics (INDIN),* pp. 440-445.

Peters, B.A., Smith, J.S., Curry, J., LaJimodiere C., 1996, Advanced Tutorial - Simulation Based Scheduling and Control, *Proceedings of the 1996 Winter Simulation Conference,* pp. 194-198.

Poslad, S., 2007, Specifying Protocols for Multi-Agent Systems Interaction, *ACM Transactions on Autonomous and Adaptive Systems,* Vol 2.

Radu, F., Frank, F., 2006, Development and applications of holonic manufacturing systems: a survey, *Journal of Intelligent Manufacturing,* Vol 17, pp. 111-113.

Ricci, A., Santi, A., 2011, Agent-oriented computing: Agents as a paradigm for computer programming and software development, *The 3rd Conference on Manufacturing Systems,* Vol 7, pp. 503–508.

Shen, W., Hao Q., Yoon, H. J., Norrie, D. H., 2006, Applications of agent - based systems in intelligent manufacturing: An updated review, *Advanced Engineering Informatics,* Vol 20, pp. 415-431.

Sturgeon, J., 2002, Modular production networks: a new American model of industrial organization, *Industrial and Corporate Change,* pp. 451- 496.

Su, J., 2007, *Component-based intelligent control architecture for reconfigurable manufacturing systems, Ph.D. Thesis,* Virginia Polytechnic Institute and State University.

Teahan, W., 2010, *Artificial Intelligence – Agent Behaviour,* BookBoon.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Petters, P., 2003. Reference Architecture for Holonic Manufacturing Systems: PROSA. *Computers in Industry,* Volume 37, p. 95–108.

Vlad, V., Graur, A., Ciufudean, C., Buzduga, C., 2010, Towards Using the Reconfiguration Capability of IEC 61499 Specifications for Modeling and Implementing Dynamic Holonic Interactions, *12th WSEAS International Conference on Automatic Control, Modelling & Simulation,* pp. 213-218.

Wang, F., Hou, Z.G., Xu, D., Tan, M., 2005, An Agent-Based Holonic Architecture for Reconfigurable Manufacturing Systems, *Advances in Natural Computation*, pp. 622-627.

Wooldridge, M.J., Nicholas, N.R., 1995, Intelligent agents: Theory and practice, *The knowledge engineering,* pp. 115-152.