Applying the PROSA Reference Architecture to Enable the Interaction between the Worker and the Industrial Robot

Case Study: One Worker Interaction with a Dual-Arm Industrial Robot

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: Flexible Manufacturing Paradigm, Holonic Manufacturing System, Multi-Agent System, Worker-Industrial Robot Cooperation, Semi-Autonomous System, Human-Robot Interaction, Gesture Recognition.

Abstract: Involving an industrial robot in a close physical interaction with the worker became quite possible, as a result of the availability of different collaborative industrial robots in the market. The physical cooperation between the industrial robot and the worker usually done under the umbrella of the flexible manufacturing paradigm, where both the industrial robot and the worker need to change their tasks fast and efficiently, to cope with the changes in the manufacturing process. This means that a reliable manufacturing control system must stand behind this physical interaction to achieve the proper communication interaction. A holonic control architecture is an ideal solution for this problem. Therefore, during this research we study the most commonly applied model of the holonic control architecture, then we apply this architecture on our case study, where one worker cooperates with a dual-arm industrial robot to build and produce any new product. Also the research uses the worker's hand gesture recognition as a method to interact with the industrial robot during the execution of a cooperative production scenario.

1 INTRODUCTION

Human-Robot Interaction (HRI) is the field of research focuses on understanding, designing and evaluating robotic systems which involve the human as an essential element of these systems (Goodrich and Schultz, 2007). The HRI fundamental rules are originally based on a fictional speculation as stated by author Isaac Asimov's novel "I, Robot" (Pinker, 1999). The first two fundamental rules of the HRI are as following:

- A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- A robot must obey orders given to it by human beings.

As a scientific interpretation of these rules. The first fundamental is mainly addressing the problem of a safe physical HRI, while the second rule addresses the problem of HRI information communication. Accordingly the term interaction in the context of HRI can either mean physical or information interaction. A very good example where the HRI physical close proximity interaction is more dominant than the information interaction is an exoskeleton robotic suit. In an exoskeleton the human literally wears a robotic suit to magnify his strength and endurance (Fontana, 2014). On the other hand, a tele-operated robot arm operates in a nuclear reactor (Parker and Draper, 1998) is an example of a remote HRI where the information interaction is more dominant than the physical interaction. However in most of the HRI scenarios, both forms of interaction coexist together, regardless which one is more obvious or dominant.

The HRI has gained a great attention in industrial applications in particular (Lasota et al, 2014). As a result of this attention, a new generation of safe cooperative industrial robots are now available in the commercial market. Example of these robots are KUKA lightweight, Rethink Baxter, YuMi ABB dual arm, and Universal Robots. These robots apply different technologies and methodologies to grantee the human safety during a close proximity interaction. Therefore, safe physical HRI is achieved. However, there is no clear control architecture which specifies the information communication interaction between

R. Sadik A. and Urban B.
Applying the PROSA Reference Architecture to Enable the Interaction between the Worker and the Industrial Robot - Case Study: One Worker Interaction with a Dual-Arm Industrial Robot.
DOI: 10.5220/0006191801900199
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 190-199
ISBN: 978-989-758-219-6

the safe industrial robot and the worker in a cooperative production scenario.

In section 2 of this paper, we introduce some well-known methodologies and terminologies which are basic keys in Intelligent Manufacturing System (IMS) design. It is important to know these methodologies and terminologies in order to formulate the problem statement in section 3. Section 4 explains the technologies which are used to apply the solution concept. Therefore an overview of the solution model can be explained in section 5. Section 6 shows a case study where we implement the solution concept. Finally section 7 wraps up the work summary with the conclusion and the future research.

2 BASIC KEY CONCEPTS

2.1 Flexible Manufacturing System

A Flexible Manufacturing System (FMS) is an integrated system of machine modules and material handling units which together can offer a certain amount of flexibility (Lozano et al, 1994). A typical FMS is composed of a series of manufacturing workcells which controlled by a stand-alone controller. A workcell can contain an automatic machine, a material handling and storage station, an industrial robot, or a human worker.

The flexibility term can be applied to different aspects of the manufacturing system, such as: the machine, the material handling and routing, the control system, and the product building plan and volume (Elmaraghy, 2005). The objectives of the FMS are to increase the manufacturing system reliability, optimizing the cycle time, reduce the lead times and costs, overcome the internal changes like breakdowns, and improve the worker productivity, the machine efficiency, and the overall quality (Koren et al, 1999; Kruger, 2015).

A Worker and Industrial Robot (W&IR) cooperative workcell is a novel case study of the FMS, where the worker can use the industrial robot as an intelligent tool to achieve the FMS objectives. The FMS control system could follow a centralized or a distributed control topology. In a centralized control topology, all the workcells controllers are supervised by one central controller which carry out the final decision making. In a distributed control topology, the decision making task is distributed between all the controllers. The scope of this research focus on distributed control approach as a modern intelligent method to control the W&IR cooperative workcell.

2.2 Holonic Manufacturing System

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 a higher level of control instructions. His conclusion was 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 IMS consortium, to define a new paradigm for the factory of the future. IMS has defined the holon as an autonomous cooperative building block of the manufacturing system, that can be used to transform, transport, store and/or validate the information and the physical objects (Radu and Frank, 2006).

The Holonic Manufacturing System (HMS) is basically a distributed control and communication topology which divides the manufacturing process tasks and responsibilities over different holon categories. Two well-known reference models are following the holonic manufacturing architecture which are: Product-Resource-Order-Staff Architecture (PROSA) model (Van Brussel et al, 2003), ADAptive holonic COntrol aRchitecture (ADACOR) model (Leitao and Restivo, 2006).

The PROSA reference model implements three basic holons as shown in Figure 1-a. These holons are resource, product, and order holons. The resource holon is a physical entity within the manufacturing system, it can represent a robot, machine, worker, etc. The product holon stores the process and the product tasks needed to insure the correct manufacturing of a certain product. An order holon is responsible for assigning the tasks and making sure they have been accomplished. The ADACOR reference model (Leitao and Restivo, 2008) implements the same three PROSA basic holons plus a fourth holon called a staff or a supervisor holon. The supervisor holon is providing coordination services when it is needed to cooperate outside the boundaries of the workcell.

The holon generic structure is shown in Figure 1-b. The resource holon is usually composed of two components which are responsible for the physical and communication interaction respectively. While the other holons can have only the communication component.

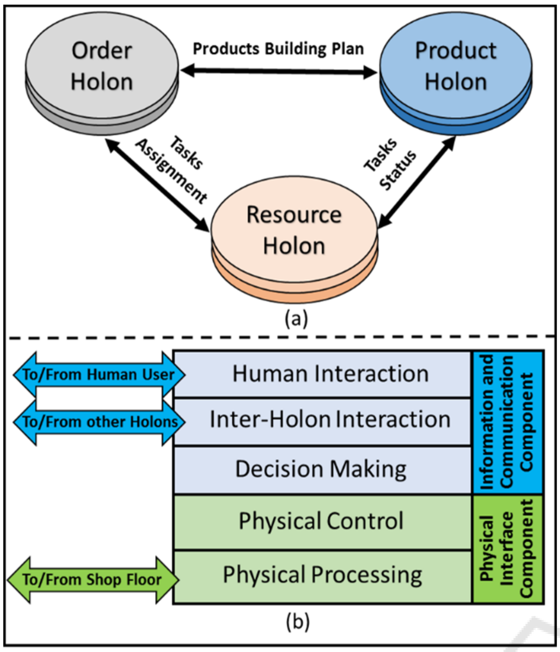


Figure 1: (a) PROSA HMS Reference Model – (b) Holon Generic Structure.

3 PROBLEM FORMULATION

Close physical interaction between the worker and the industrial robot is a new trend in manufacturing. This trend was proposed as a part of Industrie 4.0 recommendations as a new vision for the smart factory of the future. Especially for the Small and Medium-Sized Enterprises (SMEs), as they apply a highly customized industrial processes which follow the flexible manufacturing paradigm. Accordingly, the worker existence alongside the industrial robot is essential in an SME, as the worker can easily adapt to the fast changes in the production requirements. Simultaneously, the industrial robot is important resource on the factory shop floor, as it is reliable in terms of speed, load lifting, and accuracy, etc.

The first fundamental rule of the HRI can be easily obtained in a W&IR cooperative workcell, due to the existence of a variety of safe industrial robots in the commercial market. The problem is to achieve the second fundamental rule of the HIR, which is to connect the worker and the industrial robot together from the information communication point of view.

The PROSA reference model is a well-known control and communication architecture which has commonly used to solve different FMS problems. However implementing the PROSA model over this specific case of the FMS (i.e. a W&IR cooperative workcell) is a new problem. The main purpose of applying the PROSA model is to afford the worker the privilege to use the industrial robot as an intelligent tool within the production process. In a semi-autonomous control manner, the control system should allow the worker to interfere the production process and teach the robot a new task. This new task can be stored in the system and used as a building block of more complicated products.

Furthermore, in order to complete the W&IR interaction, the control system must be able to recognize the worker activities. Therefore, the second part of this research problem is to find an appropriate recognition method which can be used to express the different tasks done by the worker.

4 BASIC KEY TECHNOLOGIES

4.1 Autonomous Artificial Agent and Multi-Agent System

A software agent is a computer system situated in a specific environment that is capable of performing autonomous actions in this environment in order to meet its designer objective (Jennings & Wooldridge, 1998). An agent is responsive, proactive and social. Responsive means that the agent can perceive its environment and respond in a timely fashion to the changes occurring in it. Proactive means that the agent is able to exhibit opportunistic, goal directed behaviour and take initiative. Social means that the agent can interact with other artificial agents or humans within its environment in order to solve a problem.

Conceptually, an agent is a computing machine which is given a specific task to execute. Therefore, it chooses certain set of actions and formulate the proper plans to accomplish the assigned task. The set of actions which are available to be chosen by the agent are called a behavior. The agent behaviors are mainly created by the agent programmer. An agent can formulate one or more plan to reach its target. The selection of an execution plan among others would be based on a certain criteria which has been defined by the agent programmer. Building an execution plan is highly depending on the information which inferred by the agent from its environment. 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 et al, 2006).

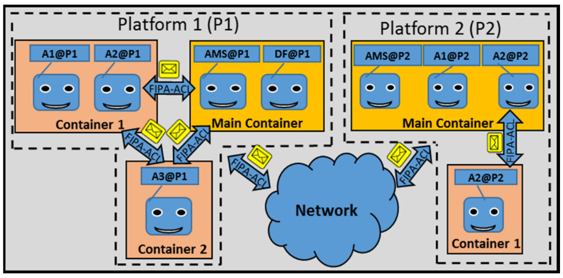


Figure 2: Java Agent Development (JADE) Framework.

JAVA Agent Development (JADE) is a distributed MAS middleware framework (JADE, n.d.). 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 (Bellifemine et al, 2007). FIPA is an IEEE computer society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies (FIPA, n.d.). JADE agents use FIPA-Agent Communication Language (FIPA-ACL) (Poslad, 2007), to exchange messages either inside its own platform or with another platform in a distributed MAS as shown in Figure 2.

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 platform, in order to facilitate agent service exchange, so that each agent can obtain its specific goal (Caire, 2009; Teahan, 2010).

4.2 Hand Gesture Recognition

One of the most successful gesture recognition techniques is the vision-based sensing that applies active technique approach. In active sensing a signal of a burst of (light, microwaves or sound) waves is emitted, then the reflected signal by the surrounding is received back by the sensor. Often the sensor is non active as long as no motion occurs in its sensing range, till some object moves within this sensing range, therefore the change in the reflected signal activates the sensor. The most famous sensors which belong to this category are the Kinect and Leap Motion (Berci and Szolgay, 2007).

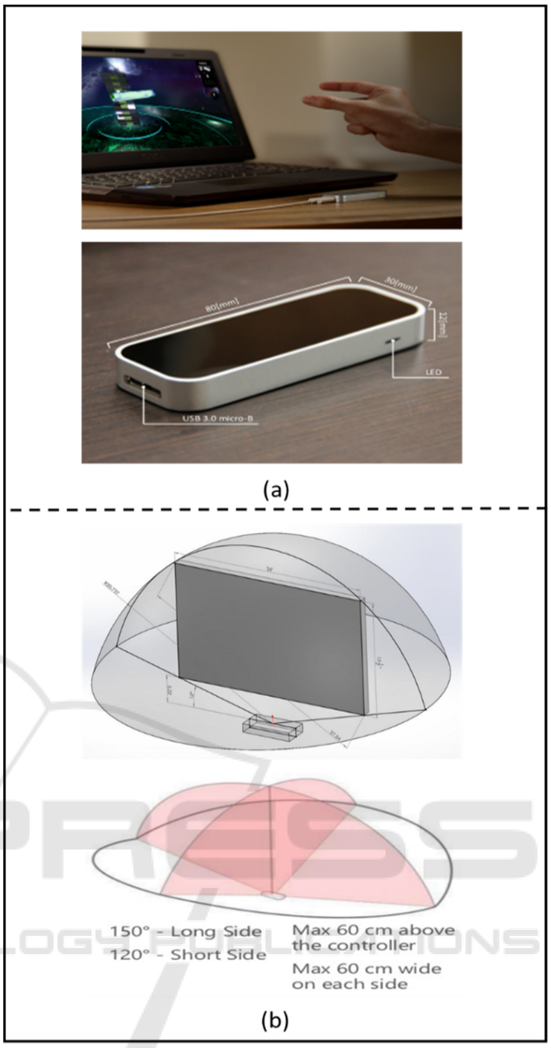


Figure 3: (a) Leap Controller Dimensions – (b) Leap Controller Coverage Range.

The Leap Motion controller (Leap Motion, n.d.) is a sensor device that aims to translate the hand movements into computer commands. The Leap dimensions are 8 cm in length and 3cm in width, and it can be connected to the computer using a USB connection as shown in Figure 3-a. The controller range of sensing is a hemispherical volume which extends to of 60 cm in radius as shown in Figure 3-b. Using two monochromatic IR cameras and three infrared LEDs the device observes its sensing volume. The infrared LEDs emit a 3D pattern of IR light dots, simultaneously the cameras reconstruct the reflected data in a rate of a 300 frames per second. The constructed data transfers to the host computer via the USB connection, where it can be parsed by the Leap Motion controller software (Potter et al, 2013).

4.3 Cooperative Industrial Robots: Baxter Rethink Robot

The robot Baxter is a collaborative dual-arm robot. Every arm is a 7 DOF articulated industrial robot. Collaborative robotics is to share the process between two robots or a robot and a human being, unlike the conventional industrial robot that repeatedly performs the very same task on the production line without a direct interaction with the human coworkers in the process. Baxter is slower than the traditional industrial robot and is "compliant", i.e. its movements are elastic, non-hazardous to the humans and closer to that found in the nature. The objective of the collaborative robotics, rather than performing a single task constantly, which is the case of a conventional industrial robot, is to be able to carry several small tasks, alongside humans in its work environment. Baxter is particularly suitable for SMEs that do not have sufficient volume to automate sophisticated tasks. But they can achieve higher productivity if they automate many simple tasks (Rethink Robotics, n.d.).

The maximum payload of one of Baxter's arm is 2.3 Kg. The two arms are programmable within the same code. Baxter has mechanisms of "anti-self-collision" for compensating its movements to avoid collisions of the two arms. These mechanisms may be deactivated in specific cases (passage of an object from one hand to the other, handling two hands close together). It is therefore possible to synchronize the two arms movements to coordinate them together or controlling any of them separately.

5 SOLUTION MODEL

While the PROSA is a conceptual model focuses on HCS description, it does not specify a certain technology to apply this concept. On the other hand, artificial agent technology is a general purpose solution which can apply the PROSA concept. Thus during this research, JADE agent framework has been used to implement the concept of the PROSA. Figure 4 illustrates the implementation of the PROSA basic holons over a W&IR cooperative workcell. With the assumption that a single worker cooperates with the Baxter dual-arms in flexible manufacturing scenario.

On the worker platform, there are four different holons, two of them are resource holons which locate on the shop floor layer. The first resource holon is the worker User Interface (UI). This worker UI holon has a physical component which is the worker laptop. The communication component of this holon is implemented in the automation layer in the form of the worker task display agent. The worker UI is basically responsible for creating the new tasks either for the worker or the robot, and constructing new products. Also the worker UI displays the assignment task for the worker during the product execution. The second resource holon is the worker task execution.

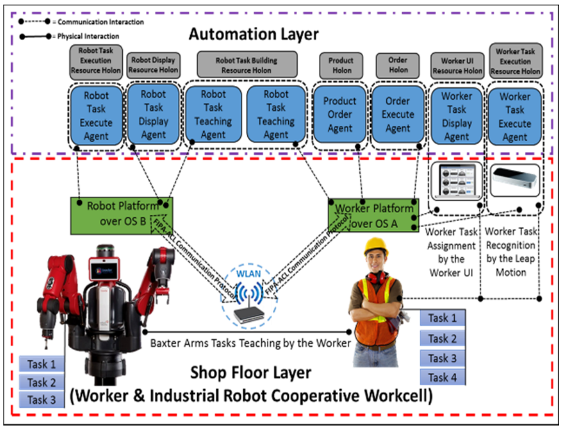


Figure 4: Worker & Industrial Robot Flexible Cooperative Scenario based on Holonic Manufacturing Concept.

The worker task execution uses the Leap controller as a physical competent to capture the worker's hand gestures which are associated with different work events. The communication component of this holon is the worker task execute agent. The last two holons on the worker side are the product and the order holons, which only implement communication components, therefore they are only located in the automation layer in a form of autonomous artificial agents.

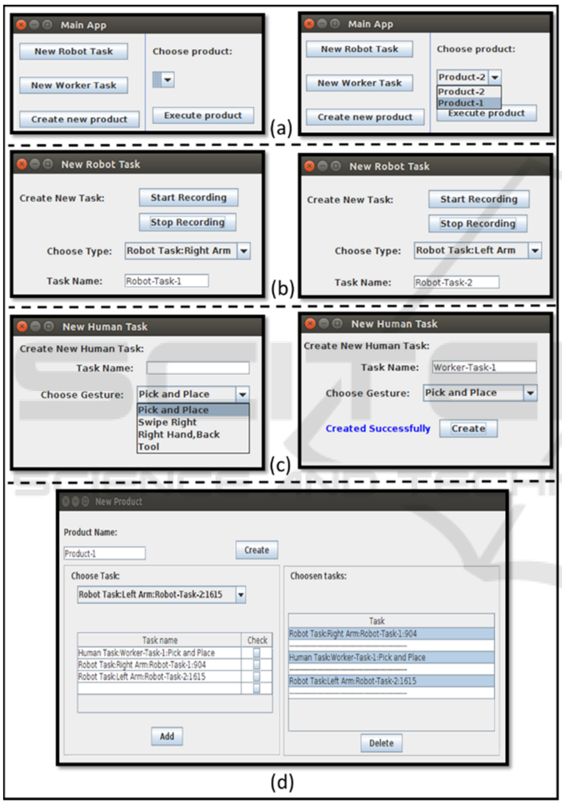
On the robot platform, there are two resource holons. The first resource holon is the robot display. Similar to the worker UI holon, the robot display holon has two components, its physical component is the Baxter screen, which is connected to the robot task display agent as its communication component. The robot display holon displays the baxter task while execution. The second resource holon is the robot task execution. The physical component of this holon could be Baxter right or left arm. In either cases the Baxter arms are connected with the robot task execution agent as their communication component. The robot task execute holon can assign a task either for the right arm, the left arm, or both.

The last holon in this solution model is the robot task building resource holon. This holon is in fact distributed over both the worker and the robot platforms. This is because it is used by the worker to teach one of the Baxter arms a new task. The worker platform and the robot platform are communicating with each other's using a Wireless Local Area

Network (WLAN) technology which deploys FIPA-ACL Communication Protocol.

6 CASE STUDY

During this section, a simple product will be built and executed using the previously mentioned solution concept. The aim is to show as simple as possible how the holons interact. Therefore, a product composed of three consequent tasks will be constructed. The first task will be assigned to the Baxter right arm, the second task will be assigned to the worker, and the final task will be assigned to the Baxter left arm.



(a) Choose product: Product-2; Product-2, Product-1

(b) Choose Type: Robot Task:Right Arm; Task Name: Robot-Task-1 — Choose Type: Robot Task:Left Arm; Task Name: Robot-Task-2

(c) Choose Gesture: Pick and Place; Pick and Place, Swipe Right, Right Hand,Back, Tool — Task Name: Worker-Task-1; Choose Gesture: Pick and Place; Created Successfully

(d) Product Name: Product-1

Choose Task: Robot Task:Left Arm:Robot-Task-2:1615

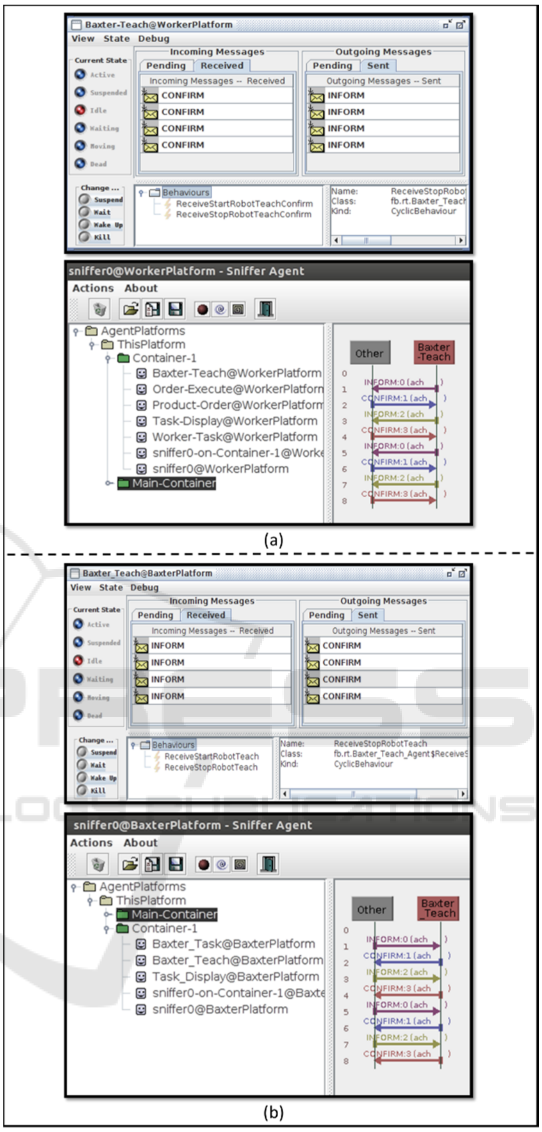
Task name	Check
Human Task:Worker-Task-1:Pick and Place	☐
Robot Task:Right Arm:Robot-Task-1:904	☐
Robot Task:Left Arm:Robot-Task-2:1615	☐
	☐

Task
Robot Task:Right Arm:Robot-Task-1:904
Human Task:Worker-Task-1:Pick and Place
Robot Task:Left Arm:Robot-Task-2:1615

Figure 5: A Case Study for a Three Tasks Product.

6.1 Product Building

Figure 5 shows the worker UI and the different steps to build a new product. Figure 5-a shows the Main App UI, which is mainly used to navigate to create a new task either for the robot or the worker, and create/execute a new product. Figure 5-b shows the creation of new robot task via the robot task UI. In our case study, two tasks are assigned to the Baxter. Before the worker starts teaching the Baxter a new task, he gives the task a name and selects which arm will be used. Then presses start recording button to physically teach the robot arm the new task, ultimately the worker presses stop recording button when finish teaching.



(a) Baxter-Teach@WorkerPlatform — Incoming Messages -- Received: CONFIRM, CONFIRM, CONFIRM, CONFIRM; Outgoing Messages -- Sent: INFORM, INFORM, INFORM, INFORM

Behaviours: ReceiveStartRobotTeachConfirm, ReceiveStopRobotTeachConfirm

Name: ReceiveStopRobot
Class: fb.rt.Baxter_Teach
Kind: CyclicBehaviour

AgentPlatforms / ThisPlatform / Container-1: Baxter-Teach@WorkerPlatform, Order-Execute@WorkerPlatform, Product-Order@WorkerPlatform, Task-Display@WorkerPlatform, Worker-Task@WorkerPlatform, sniffer0-on-Container-1@Worke, sniffer0@WorkerPlatform; Main-Container

0
1 INFORM:0 (ach)
2 CONFIRM:1 (ach)
3 INFORM:2 (ach)
4 CONFIRM:3 (ach)
5 INFORM:0 (ach)
6 CONFIRM:1 (ach)
7 INFORM:2 (ach)
8 CONFIRM:3 (ach)

(b) Baxter_Teach@BaxterPlatform — Incoming Messages -- Received: INFORM, INFORM, INFORM, INFORM; Outgoing Messages -- Sent: CONFIRM, CONFIRM, CONFIRM, CONFIRM

Behaviours: ReceiveStartRobotTeach, ReceiveStopRobotTeach

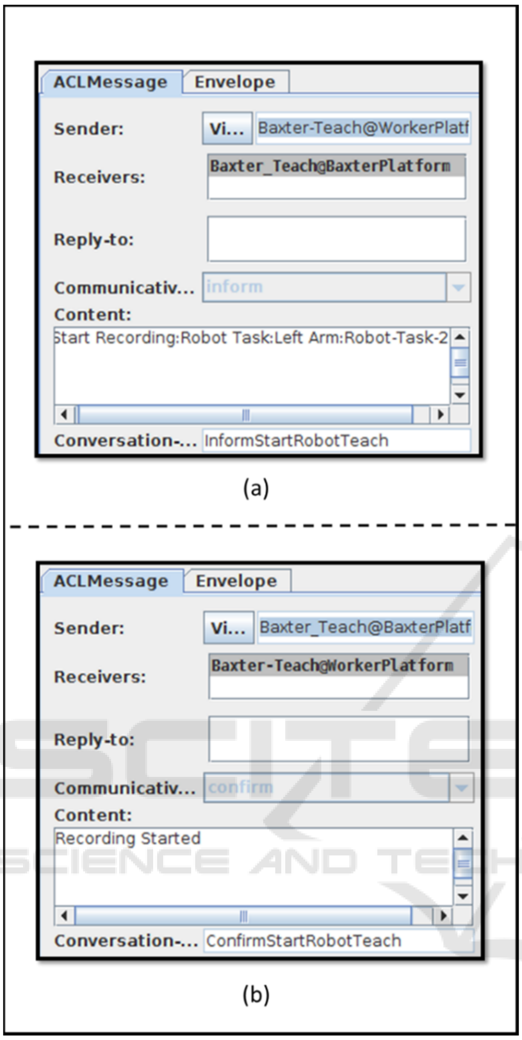
Name: ReceiveStopRobotTeach
Class: fb.rt.Baxter_Teach_Agent$ReceiveS
Kind: CyclicBehaviour

AgentPlatforms / ThisPlatform: Main-Container; Container-1: Baxter_Task@BaxterPlatform, Baxter_Teach@BaxterPlatform, Task_Display@BaxterPlatform, sniffer0-on-Container-1@Baxte, sniffer0@BaxterPlatform

0
1 INFORM:0 (ach)
2 CONFIRM:1 (ach)
3 INFORM:2 (ach)
4 CONFIRM:3 (ach)
5 INFORM:0 (ach)
6 CONFIRM:1 (ach)
7 INFORM:2 (ach)
8 CONFIRM:3 (ach)

Figure 6: JADE Interaction to Teach a New Task for the Baxter.

Figure 6 shows the communication interaction between Baxter-Teach@WorkerPlatform and Baxter_Teach@BaxterPlatform in order to teach Baxter a new task. An INFORM message is sent from the worker side in case of start/stop recording, and the message is replied by a CONFIRM message from the Baxter side to complete the handshaking process as it can be seen in Figure 7. The start teaching message holds the name of the task and which arm should perform this task. Accordingly the task will be recorded on Baxter Platform using this exact task name as a reference.



Sender: Vi... Baxter-Teach@WorkerPlatf

Receivers: Baxter_Teach@BaxterPlatform

Reply-to:

Communicativ... inform

Content:
Start Recording:Robot Task:Left Arm:Robot-Task-2

Conversation-... InformStartRobotTeach

(a)

Sender: Vi... Baxter_Teach@BaxterPlatf

Receivers: Baxter-Teach@WorkerPlatform

Reply-to:

Communicativ... confirm

Content:
Recording Started

Conversation-... ConfirmStartRobotTeach

(b)

Figure 7: ACL-Messages exchange to start task recording.

Figure 5-c shows the creation of a new worker task via the worker task UI. In our case study four different gestures have been programmed to be recognized by the Leap Motion Controller. Those four gestures are Pick and Place an object, Horizontal Swipe the hand from left to right, Lean the right hand back, and finally a Tool (Screw Driver) recognition. Swipe hand and Tool are pre-defined gestures by the Leap SDK. However Pick and Place and Lean right hand back are customized gestures by our code which will be shown in the next section. Pick and Place has been selected as an indication of the worker task. After defining all the needed tasks from both the robot and the worker to build a product. A product building plan can be defined using the product UI as shown in Figure 5-d.

6.2 Product Execution

After pressing execute product button in the Main App UI, an ACL-Message with a communicative act "AGREE" is sent from the Product Order agent to the Order Execute agent as shown in Figure 8.

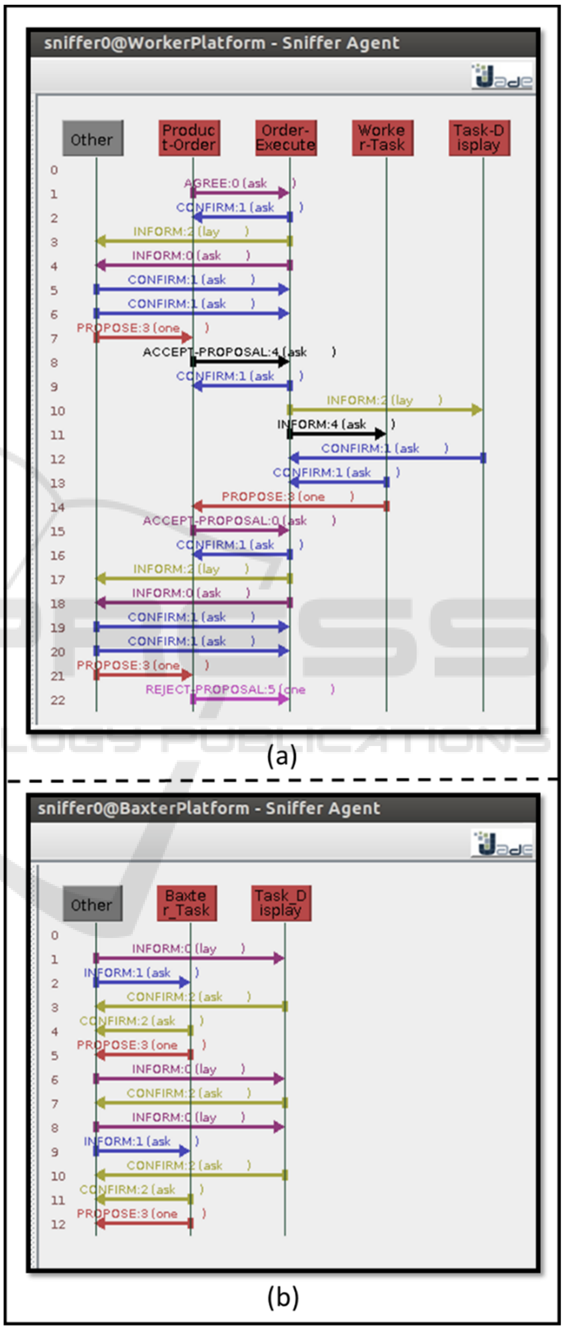
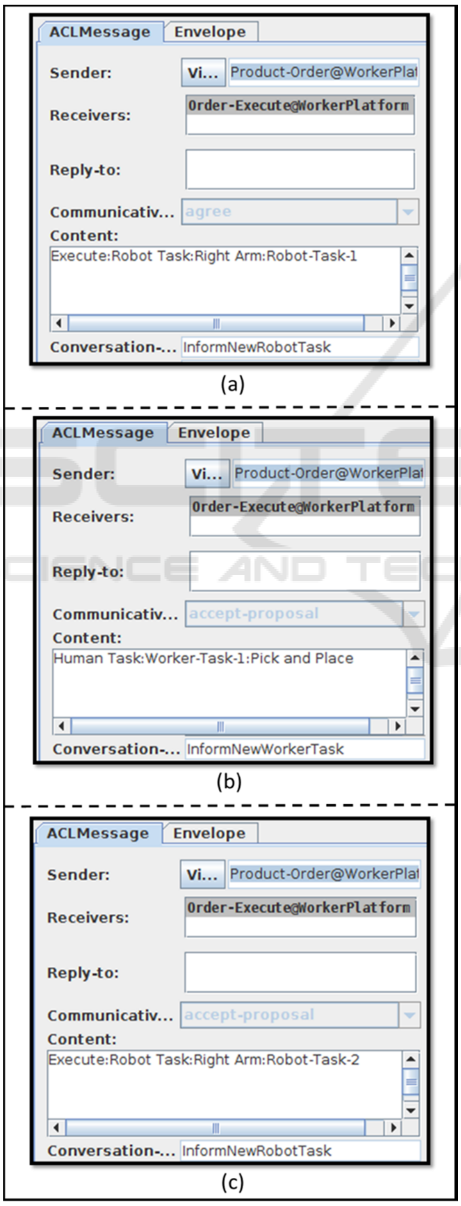


Figure 8: JADE Interaction to execute a Three Tasks Product.

The AGREE message is an indication to start the execution of a new product. The message content holds the task target and the task name as shown in Figure 9-a. the Order Execute agent answers the Product Order agent with a CONFIRM message the same way described previously in section 6.1 during teaching Baxter a new task. Based on the task target within the AGREE message, the Order Execute agent assigns a new task execution for that target. Therefore it informs the task to the Baxter_Task@BaxterPlatform and simultaneously to the Task_Dispaly@BaxterPlatform.



Sender: Product-Order@WorkerPlat
Receivers: Order-Execute@WorkerPlatform
Reply-to:
Communicativ... agree
Content:
Execute:Robot Task:Right Arm:Robot-Task-1
Conversation-... InformNewRobotTask

(a)

Sender: Product-Order@WorkerPlat
Receivers: Order-Execute@WorkerPlatform
Reply-to:
Communicativ... accept-proposal
Content:
Human Task:Worker-Task-1:Pick and Place
Conversation-... InformNewWorkerTask

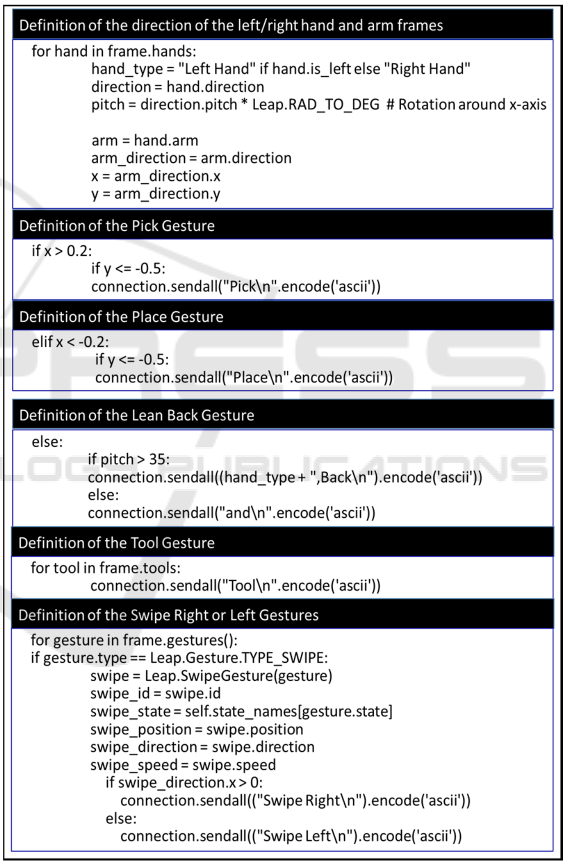
(b)

Sender: Product-Order@WorkerPlat
Receivers: Order-Execute@WorkerPlatform
Reply-to:
Communicativ... accept-proposal
Content:
Execute:Robot Task:Right Arm:Robot-Task-2
Conversation-... InformNewRobotTask

(c)

Figure 9: ACL-Messages to start tasks execution.

Both the last mentioned agents send back confirmations that they received a new task to the Order Execute agent. When Baxter Task agent receives an INFORM message to start executing a new task, it searches into the message contents for the task name. When Baxter Task agent finds a match of the task name with the tasks previously recorded. It starts to play back this exact task, and finally sends an ACL-Message with communicative act "PROPOSE" to the Product Order agent. The PROPOSE message indicates that a task done from the product. As the product tasks still not finished, the Product Order agent sends an ACL-Message with communicative act "ACCEPT-PROPOSAL" to the Order Execute agent.



Definition of the direction of the left/right hand and arm frames

```
for hand in frame.hands:
        hand_type = "Left Hand" if hand.is_left else "Right Hand"
        direction = hand.direction
        pitch = direction.pitch * Leap.RAD_TO_DEG  # Rotation around x-axis

        arm = hand.arm
        arm_direction = arm.direction
        x = arm_direction.x
        y = arm_direction.y
```

Definition of the Pick Gesture

```
if x > 0.2:
        if y <= -0.5:
        connection.sendall("Pick\n".encode('ascii'))
```

Definition of the Place Gesture

```
elif x < -0.2:
        if y <= -0.5:
        connection.sendall("Place\n".encode('ascii'))
```

Definition of the Lean Back Gesture

```
else:
        if pitch > 35:
        connection.sendall((hand_type + ",Back\n").encode('ascii'))
        else:
        connection.sendall("and\n".encode('ascii'))
```

Definition of the Tool Gesture

```
for tool in frame.tools:
        connection.sendall("Tool\n".encode('ascii'))
```

Definition of the Swipe Right or Left Gestures

```
for gesture in frame.gestures():
if gesture.type == Leap.Gesture.TYPE_SWIPE:
        swipe = Leap.SwipeGesture(gesture)
        swipe_id = swipe.id
        swipe_state = self.state_names[gesture.state]
        swipe_position = swipe.position
        swipe_direction = swipe.direction
        swipe_speed = swipe.speed
          if swipe_direction.x > 0:
            connection.sendall(("Swipe Right\n").encode('ascii'))
          else:
            connection.sendall(("Swipe Left\n").encode('ascii'))
```

Figure 10: The Python Server Code to define the Worker Hand Gestures.

The ACCEPT-PROPOSAL message that has been sent to the Order Execute Agent is an indication of a new task assignment. The same previously mentioned mechanism to assign a new task for the robot is followed to assign a worker task. As this time the ACCEPT-PROPOSAL message content indicates a Worker Task with type Pick and Place as shown in Figure 9-b. Thus the Order Execute agent informs the task to the Worker-Task@WorkerPlatform and to the Task-Dispaly@WorkerPlatform. Both the last mentioned agents send back confirmations that they received a new task to the Order Execute agent. When the Worker Task agent receives an INFORM message to start executing a new task, it extracts the task name, initialized a unique socket, and waits the Leap server to capture this specific gesture. The definitions of the different gestures are written in the Leap Server using Python language as shown in the Figure 10.

When the Leap server captures the specified gesture by the Worker Task agent, it informs the Worker Task agent that this gesture was detected. Accordingly the Worker Task agent turns off the connection socket and sends a PROPOSE message to the Product Order agent to inform a task done event. The Baxter Task-2 is achieved exactly the same way Baxter Task-1 done as can be seen in Figure 9-c, except one important difference. At the final step of the product execution when the Baxter Task agent sends a PROPOSE message to the Product Order agent. The Product Order agent realizes that all the product tasks have been fulfilled, therefore it sends the Order Execute agent an ACL-Message with a communicative act "REJECT-PROPOSAL", to indicate the end of this routine.

7 SUMMARY, CONCLUSION AND FUTURE WORK

During this research, we introduced the basic fundamental rules of the HRI. In the field of industrial robots, the first fundamental rule has become easily achievable, because of the availability of safe cooperative industrial robots in the robotics market. However to achieve the second HRI fundamental rule, an intelligent control and communication architecture must be found and implemented. The purpose of this control architecture not only to connect the industrial robot and the worker together from the information point of view, but also to enable the flexible cooperation between them.

The research has applied the PROSA holonic control reference model to solve the addressed problem. The PROSA reference model specifies the three basic holons which are required in intelligent manufacturing scenarios. That was the reason of choosing this model specifically, as the scale of our study still limited to one worker in cooperation with one or two robots.

Based on the PROSA specifications, the physical and information interaction model between a worker and a dual-arm has been constructed. JADE autonomous agent framework has been used to implement and deploy the communication components of the PROSA holons. A robot task teaching resource holon has been used by the worker to teach any of the Baxter arms a new task. While a robot task execution resource holon was responsible for executing this task. The Leap sensor has been chosen to track the gestures of the worker's hand, accordingly a dedicated resource holon was assigned to monitor the Leap Motion controller server and pass the gestures to this resource holon when it is needed.

As has been shown in the case study, the production flexibility has been accomplished using the proposed solution. Thus, the worker could teach the robot many different tasks at any time during the manufacturing process. Simultaneously, he can assign different hand gestures which are associated with his activities during the production process. Ultimately, the worker can use different combinations of his hand gestures along with the robot tasks which are stored in the system to construct more complicated production routines. Accordingly, the proposed solution can increase the manufacturing system reliability by adapting to the production requirements, reduce the final production lead time and cost, and improve the worker productivity and the robot efficiency.

As the coordination between two or more than worker and industrial robot is out of context of this research, a staff holon has not been implemented. However, in the future work the proposed solution can be scaled over more than one worker and industrial robot. Therefore the staff holon can be taken into consideration. Also during the future work, more hand gestures can be investigated and studied, thus it can be offered to the worker to build more sophisticated products.

REFERENCES

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

Berci, N., Szolgay. P., Vision based human–machine interface via hand gestures, *18th European conference on circuit theory and design (ECCTD 2007)*, pp 496–499, 2007.

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

Elmaraghy, Hoda A., Flexible and reconfigurable manufacturing systems paradigms, *International Journal of Flexible Manufacturing Systems*, Volume 17, Issue 4, pp 261-276, October 2005.

FIPA, Fipa.org, 2016. [Online]. Available: http://www.fipa.org/. [Accessed: 06- Oct- 2016].

Fontana, M., the Body Extender: A Full-Body Exoskeleton for the Transport and Handling of Heavy Loads, *IEEE Robotics & Automation Magazine*, Vol. 21, pp. 34-44, 2014.

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

Goodrich, M.A., Schultz, A.C., Human–robot interaction: a survey, *Foundations and Trends in Human–Computer Interaction* Vol. 1, pp. 203–275, 2007.

JADE | Java Agent DEvelopment Framework, *Jade.tilab.com*. [Online]. Available: http://jade.tilab.com/. [Accessed: 1- Oct- 2016].

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

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

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.

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

Leap Motion, Leapmotion.com, [Online]. Available: https://www.leapmotion.com/. [Accessed: 03- Oct-2016].

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

Leitao, P., Restivo, F., Implementation of a Holonic Control System in a Flexible Manufacturing, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS,* Vol. 38, pp. 699-709, 2008.

Lozano, J., Teba, J., Larrañeta, L., Álvarez, P., Dynamic part-routing in a Flexible Manufacturing*, Journal of Operations Reaserch, Statistics and Computer Science*, Vol. 34, pp. 16-28, 1994.

Parker, L.E., Draper, J.V, Robotics Applications in Maintenance and Repair, *Handbook of Industrial Robotics*, 2nd Edition, J. Wiley, 1998.

Pinker, S., How the Mind Works, *Annals of the New York Academy of Sciences*, Vol. 882, pp. 119–127, 1999.

Potter, L., Araullo, J., and Carter, L., "The Leap Motion controller: A view on sign language," in *25th Australian Computer-Human Interaction Conference*, 2013.

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

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

Rethink Robotics | Advanced Robotics Technology | Collaborative Robots, *Rethink Robotics*, [Online]. Available: http://www.rethinkrobotics.com/. [Accessed: 02- Oct- 2016].

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

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

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

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