Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 37 (2015) 254 – 259

www.elsevier.com/locate/procedia

CIRPe 2015 - Understanding the life cycle implications of manufacturing

ROS based coordination of human robot cooperative assembly tasks-An industrial case study

Panagiota Tsarouchi[a], Sotiris Makris[a], George Michalos[a], Alexandros-Stereos Matthaiakis[a], Xenofon Chatzigeorgiou[a], Athanasios Athanasatos[a], Michael Stefos[a], Panagiotis Aivaliotis[a], George Chryssolouris[a] *

[a]*University of Patras, Department of Mechanical Engineering and Aeronautics, Laboratory for Manufacturing Systems and Automation, Patras 26500, Greece*

* Corresponding author. Tel: +30-2610-997262; Fax: +30-2610-997744. *E-mail address:* xrisol@lms.mech.upatras.gr

Abstract

This paper discusses a method for the coordination of assembly tasks requiring the cooperation of humans with a robot. A ROS based architecture is used. The assembly sequence of these tasks and their characteristics are modeled in a neutral XML format generated off-line. A ROS based framework in order for different modules to communicate and coordinate their actions through the exchange of messages.. The human is able to review the past and upcoming tasks in a graphical user interface. The human and robot coexist in a fenceless cell where safety is ensured using a certified camera system. This framework is applied to an automotive case study using the Process Simulate tool for the execution of assembly tasks.

Peer-review under responsibility of the organizing committee of CIRPe 2015 - Understanding the life cycle implications of manufacturing

Keywords: Human Robot cooperation; ROS; Assembly.

1. Introduction

Human robot cooperation is a promising way of achieving better productivity and flexibility [1] . Programming of industrial robots involve [2] online programming via teach pendants and Off-Line Programming (OLP) tools [3]. A three-dimensional layout of a cell is used for the creation of efficient robot paths in OLP tools. Thus, the time for online robot programming and path refinements is decreased. Despite the fact that Offline programming is widely used in automotive industry [4], digital human models (DHMs) are not always integrated [5]. The Process Simulate from Siemens PLM, 2014 (Jack model), DELMIA from 3DS, 3D Automate from Visual Components etc. are some platforms that have already integrated human models.

This paper presents a method that uses OLP data for human and robot tasks. The coordination of human and robot tasks is also possible without using software modules for robot-specific programs generation. To this effect, a software framework, based on the Robot Operating System (ROS) [26] is proposed as a middleware. The interaction between different robot controllers and OLP tools is possible through this framework. The simulation data are generated in a neutral XML format. A 'command handler' converts the XML file data into the robot maker language. This method enables human integration in a hybrid assembly cell and the coordination of assembly tasks sequence.

There are examples of existing robotic cells, where collaboration between a robot and human takes place [6][7][8][9][10][11]. An extended review on Human-aware robot navigation is presented in [12]. An algorithm for online robot path adaptation and optimization is proposed, combining input from human. A method for task allocation between a human and a robot using intelligent planning techniques has been discussed in [13]. For the efficient coordination of human and robot tasks, numerous research activities have focused on dynamic planning techniques. In this way, the dynamic allocation of tasks to humans and

Peer-review under responsibility of the organizing committee of CIRPe 2015 - Understanding the life cycle implications of manufacturing
doi:10.1016/j.procir.2015.08.045

robots is allowed. The dynamic scheduling and re-scheduling techniques are essential due to two main reasons. Firstly, the humans' actions are not always deterministic and secondly, these actions are sometimes different from the decision-making authority humans are given. For such a method an approach based on semantic maps was proposed in [14]. This method focused on making the agents more autonomous by improving the use of generic knowledge.

The communication and synchronization among software modules is directly connected to the coordination of actions [15]. The service oriented architectures (SOA) have already been used as a computing paradigm. There are already some robotics applications that have adopted SOA approaches [16], [17]. They still require high computing time and are not characterized as general purpose platforms. These are the main reasons for which they are not widely adopted [18]. The communication among different software modules that are loosely coupled through the exchange of messages is SOA framework main advantage. In the case of human-robot cooperative tasks, a SOA approach can help in the online tasks' allocation, based on the feedback from sensors [19]. In addition, the SOA frameworks provide easier interaction among hardware resources such as robot and sensors. Thus, the focus is given to the software developments, rather than the hardware integration details (e.g. use of PLCs for managing I/Os) [20].

In the SOA frameworks, the communication between the hardware resources is achieved through software applications interaction. The related data are stored into data repositories rather than into the hardware resource control. This enables the Machine-to-Cloud (M2C) communication [21]. The main advantage of such systems in robotics applications is the reduction of setup time, as the data are not managed by the resource control. Major changes in a hardware module can be achieved through the application of different middleware. This also enables the extensibility and modularity of the control system.

This paper deals with the coordination of collaborative tasks. Physical aspects, namely the parts' weight, for the allocation of collaborative tasks have not been considered. Section 2 describes the assembly sequence of human-robot cooperative tasks and the data exported from the OLP tool. Section 3 presents the graphical interfaces and the architecture of human robot tasks coordination. Section 4 deals with the fenceless human robot cooperative cell and the safety related aspects. The automotive industry case study is described in Section 5.

2. Assembly sequence of Human Robot tasks

The assembly sequence of human and robot tasks is based on the hierarchical model described in [22]. Three different levels are deployed; namely order, tasks and operations. The tasks include a number of operations that can be executed by a human, a robot or both. In the operation level, the human operations are referred to a number of steps without any arguments, whilst the robot operations include a number of arguments, such as robot positions.

A tool has been developed for the export of human and robot tasks data, in a neutral format from an OLP tool, in order to be coordinately executed through the HR tasks coordination module. This module includes the 'command handler' application, the database and the graphical user interfaces (GUIs) that are bridged over the ROS middleware. The GUIs were able to read the XML files from the XML repository and store them into the database. These data can be loaded, using the name of the model stored into the database repository and correspond to the assembly program, containing the robot and human tasks. These data are sent to a virtual or real robot controller via a 'command handler' application. Both the human and the robot execute these tasks, in a fenceless workspace, as it is discussed in section 4.

The structure of this tool is illustrated in Fig. 1.Two main modules are also visualized in this figure. The OLP tool, which is Process Simulate (PS) [23], is used for the simulation of the assembly scenario, including human and robot tasks. A digital human model (DHM) is used for the simulation of the human tasks, on the basis of the Jack model. The program is structured by the user in the hierarchical sequence described above. Single as well as multi-arm systems can be simulated with the use of PS. In this paper, a COMAU dual arm robot is simulated. When the simulation of tasks has been completed, the user may generate their sequence in the robot manufacturer language, using a module being offered by the OLP tools providers at a specific cost. In the proposed methodology, the OLP data are exported in a neutral format, such as XML, and stored into the XML files repository. The XML file has a standard structure including information on the assembly sequence such as task, operations, next operations, resources, etc.

The system's second main module is the 'HR tasks coordination'. The XML files stored into the XML repository are accessed through GUIs for loading and storing the files data into a Database. This Database communicates with the GUIs through the exchange of ROS messages. The main functionalities of this communication include retrieving, visualizing and sending data to the robot controller through the 'command handler' application and the TCP/IP communication protocol. The communication is achieved through ROS nodes that define a pair of messages: one for the request and one for the reply. Examples of the server and client services are further explained in Section 3. The 'command handler' application that runs in the Linux PC coordinates the human and robot task execution. When a robot or a human task has been finished, a message is sent to the 'command handler' (C++) through the TCP/IP socket. This is also visualized in the GUIs, using the ROS message has been sent through the ROS middleware.

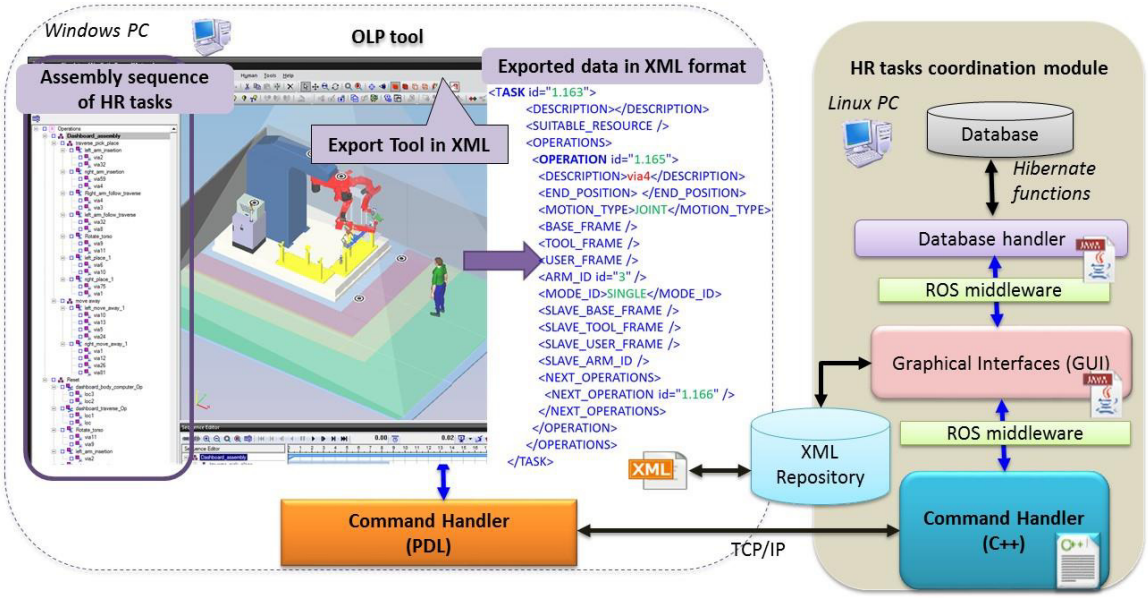

Fig. 1. Software system architecture.

The ‘Graphical Interfaces’ access to Human-Robot (HR) tasks data is achieved through the ‘Database handler’ application. This module has been implemented in Java and the ROS middleware. The visualization of the HR tasks follows this step in the Graphical Interfaces. All the information about the HR tasks are sent to the robot controller by the interaction of the two ‘command handlers’; one in the robot side and one in the HR tasks coordination module side, .The second module is independent from the robot manufacturer. Extending this application in a real world assembly cell, the PDL2 ‘command handler’ may run on a COMAU C5G robot controller.

3. Human-Robot tasks coordination module

The ‘HR tasks coordination’ module includes three main applications; the GUIs, the command and database handler. A set of Graphical Users Interfaces (GUIs) were developed for the visualization and monitoring of the HR tasks. A number of functionalities are offered through these GUIs. Firstly, the XML files exported from the OLP tool can be loaded and their data can be imported into the database. This functionality is illustrated in point 1 of Fig. 2. The human can also select and load a program that has already been stored into the database. An interface for the visualization of the hierarchical model was also designed. This interface is used both for the visualization as well as for the monitoring of the selected program’s execution, as shown in Fig. 2 (point 3). In addition, a popup graphical interface, showing messages to the human operator, appears when a human task is about to start. Moreover, this popup window enables the human to declare that he has finished his task and thus it is the robot’s turn to follow up with the rest tasks.

The selected program can be visualized in the main window of this interface. The hierarchical model used is described in section 2 and the three levels, namely the Order, Task and Operation are visualized. At the second level, there are two categories of tasks. The first category refers to the tasks assigned to the robot and the second refers to the tasks assigned to the human.

The operations of a robot task, can be either single, cooperative or synchronized motions. In the case of cooperative motions, both of the robot arms move with a master arm in order to set a motion path and the slave arm to follow. In the meanwhile, during a synchronized motion, both arms move independently but simultaneously. When a robot motion-operation is executed the user can monitor it through the GUI. In the aforementioned hierarchical tree chart, each task and operation that is currently executed is highlighted for its own duration. When a task is assigned to a human operator, the user is informed with a popup window and the human indicates the beginning of the task.

The sequence of the tasks and operations is defined during the Off-line programming. This sequence is stored into the XML file, with the use of pre and post conditions between the tasks and the operations. Two special tables for the pre and post condition are included in the database, where the XML files data are stored.

When the user selects a program stored into the database, the sequence of tasks and operations is fully retrieved and controlled by the database handler application (Fig. 2, point 2). This handler sends the information, regarding the sequence of assembly tasks to the ‘Graphical Interface’ and allows its visualization in a three level hierarchical tree, as shown in Fig. 2 point 3. The arguments of each operation following this sequence are sent step by step to the ‘command handler’ application via the ROS middleware application. The ROS Services that have been developed for the exchange of messages are summarized in Table 1.

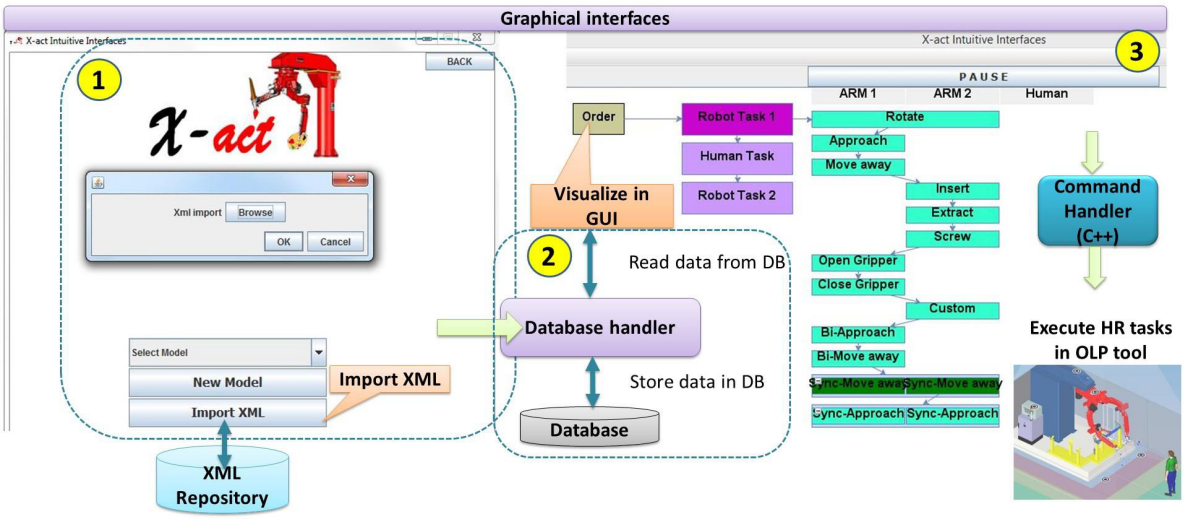

Fig. 2. Graphical interfaces for HR tasks coordination

Table 1. Services and ROS messages exchange

Communication between	Service format
Database handler- GUIs	ros::ServiceClient storeinDB (String, String)
Database handler-GUIs	ros::ServiceServer readfromDB (String Integer32, String[])
Command handler (C++) - GUIs	ros::ServiceClient executeHRtask (String, Boolean)

The above ROS messages, in the format of request/reply, coordinate the data exchange between the database and the 'command handler', in order for the correct assembly sequence to be kept. The aforementioned tool runs on an external Linux based PC with Ubuntu 12.04 Precise, and ROS Groovy [27]version.

4. Fenceless human robot cooperation

The assembly cell in this paper is fenceless, due to human robot coexistence. This is opposed to the safety standards (e.g. ISO 10218) for robots where almost all industrial robotic cells are 'caged' within fences. For this reason, a certified way that would provide safety to human operators is necessary. The only certified device, available in the market, is the SafetyEye from PILZ [24] that was installed in the cell and was connected to the robot, instead of the fences.

The SafetyEye is basically a 3D industrial camera, mounted on the ceiling above, overlooking the cell and the surrounding area. Using the markers placed on the floor, the user is able, to demarcate the area into zones with different functions using the provided software. These zones are Red (Detection Zones), or Yellow (Warning Zones). Detection Zones are the ones typically placed closer to the robot. A signal from the Safety Eye's controller is sent to the robot's controller, activating and deactivating the robot's I/O. A Safety Emergency Alarm is then triggered in the robot and the robot driver motors are switched off.

However, resetting the robot every time a Detection Zone is violated, is not practical. In order for that to be avoided, a Warning Zone is defined around the Detection Zone. This zone is used to warning the approaching operator, by turning on a yellow light indicator, informing the human about his proximity to the robot. Furthermore, because the robotic cell is surrounded by the Detection Zone, the human operator is unable to perform a collaborative task when required. Two -or more- different zone arrangements can be defined in order for that problem to be addressed. In that way, when the robot executes a program, it is completely isolated, but when the human performs a cooperative task, there is a different zone arrangement that enables the human operator to approach and complete the task.

When the human operator starts a task, the zone arrangements are switched on by a button, and when the task has been completed they are switched off. The robot will continue the program's execution without any human being around. The human operator is able to share a workspace with the robot by switching between zones. The robot tasks that are executed without human cooperation are connected to the Detection and Warning zones that allow stopping the robot when a human approaches. If a human enters the robot workspace, the configuration of the zones changes. The implementation of the fenceless human robot cooperation in a virtual environment has been achieved with the use of the virtual I/Os activation, based on the Detection and Warning zones design.

5. Automotive case study

This section presents a case study stemming from the automotive industry's manual assembly lines and is related to the assembly of a vehicle traverse. The potential of partly automating the assembly operations in this station was examined in [25]. The same case study in that paper will be executed using the tool described in the previous sections in a

simulation environment. The Graphical interfaces run on a Linux based PC that communicates with the Windows based PC, where the OLP tool runs.

The data are stored into the database through the database handler and the execution of the robot and human tasks starts (Fig. 3). The first two pictures marked with "R" present the robot task and refer to the transport of a vehicle traverse from a loading to an assembly area, as well as the installation of a Fuse Box. As shown on the floor of the cell, for these two tasks, the safety zone arrangement is dedicated to excluding the human presence, since with the red (Detection) zone the robot stops in real time.

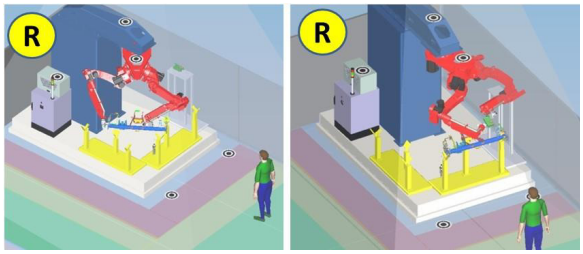

Fig. 3. Robot (R) tasks execution

The pictures marked with "H" in Fig. 4, present the task performed by the human operator, in collaboration with the robot, and include the installation of a cable, used to connecting the Fuse Box with the rest of the vehicle's components. The safety zones arrangement has been changed in order to allow to human operator to enter the common workspace and is marked as a yellow (Warning) zone.

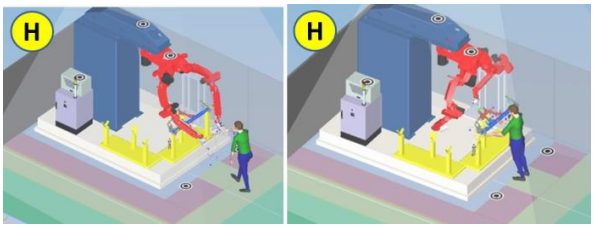

Fig. 4. Human (H) tasks execution and safe HR coexistence.

In Fig. 5, the graphical interface is presented indicating some examples of Robot 'R' and Human 'H' tasks. While the Robot 'R' task is executed, there is a presentation of the list of operations.

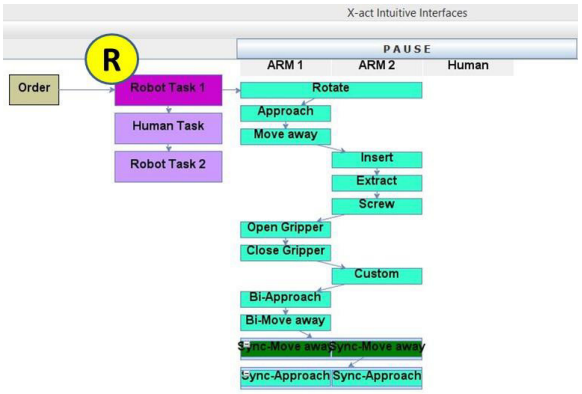

Fig. 5. Monitoring of Robot tasks under three level hierarchical model.

The operation being executed at the moment is highlighted with green, in order to enable the monitoring of execution. The names of operations are used to defining their functionality, so that the user can monitor and control the executed program with higher level of information. The tasks performed by the human operator are presented in the "H" section of Fig. 4.

Similar to the robot tasks, the operations are presented in a list and are appropriately highlighted. Since the human tasks are not well defined from a timing point of view, a popup menu automatically appears when a human task starts ('H'), as shown in Fig. 6. The human operator selects the 'START' button in order to signal the start of a human task and the 'STOP' button in order to define its completion. In this way, the next task is coordinated by the human feedback according to this information. When the Robot and Human tasks have been completed, the user may repeat the selected program execution or start the execution of a new one. The data of this program will remain in the database until the user has selected to remove the model's name.

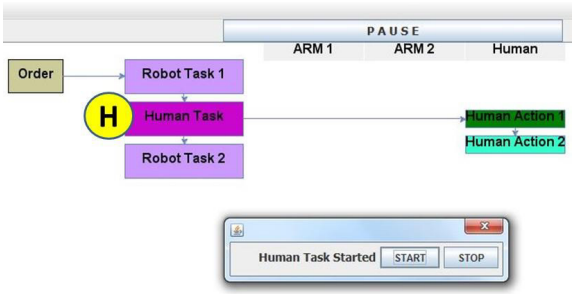

Fig. 6. Monitoring of Human tasks and Interaction during tasks execution.

6. Results & discussions

The use of OLP tools is quite popular amongst several industrial fields and especially in the automotive industry. Such a software enables the user to preliminarily design and simulate an industrial cell, before the setup in the production environment. Additionally, the initial robot programming can be carried out by the OLP tools and be translated into the robot specific language for its final tuning.

This study has proposed that the OLP data be exported in a neutral XML format, by overriding the need for different export tools of the different robot programming languages. In this way, the user is able to use the exported data without translating them into the robot controller programming language.

This approach can be applied to industrial environments, where different kinds of robots are available in the shop-floor. Furthermore, it is a good option for SMEs, as the use of customized exporters in the OLP tools is not cost efficient. Last but not least, with the proposed method, all the data used by robots are stored into external databases. Thus, the possibility of checking, modifying, processing, monitoring and structuring the robot data, even from a distance, is a major advantage.

More advantages of the method presented derive from the fact that the files exported from the OLP software are saved in a database. In that way, the files can be directly linked with the SOA framework. This facilitates the use of sensors for the execution of programming or controlling the robot, as well as

for the integration of the Human Task, as presented in the Case Study.

7. Future work

The proposed method has been applied to a virtual hybrid assembly cell, where a human operator and a dual arm robot share assembly tasks and workspaces. The next step of this framework is to be extended to a real world cell, allowing for the coordination of human robot tasks in near real industrial environments. The software modules should also be adapted to the hardware components integration, such as the integration of depth sensors for recognizing the human intention. The stability of the proposed method, as well as the acceptance of the fenceless supervision cell by the operators will also be validated in a future work. Additionally, the human robot tasks scheduling is out of this paper's scope and is a part of future research. Physical aspects as well as the robot and the human skills should be considered in the collaborative tasks scheduling.

Acknowledgement

This study was partially supported by the project X-act/ FoF-ICT-314355 (http://www.xact-project.eu/), funded by the European Commission in the 7th Framework Programme.

References

[1] Chryssolouris G. Manufacturing Systems: Theory and Practice. 2nd ed. New York: Springer-Verlag; 2006.
[2] Bottazzi V, Fonseca J. Off-line programming industrial robots based in the information extracted from neutral files generated by the commercial CAD tools. Literatur Verlag 2006
[3] Pan Z, Polden J, Larkin N, Van Duin S, Norrish J. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing 2012; 28/2:87–94.
[4] Rooks BW. Offline programming: a success for the automotive industry. Industrial Robot: An International Journal 1997; 24/1:30–34.
[5] Mourtzis D, Doukas M, Bernidaki D. Simulation in Manufacturing: Review and Challenges. Procedia CIRP 2014; 25: 213–29.
[6] Akella P, Peshkin M, Colgate E, Wannasuphoprasit W, Nagesht N, Wells J, Holland S, Pearson T, Peacockt B. Cobots for the Automobile Assembly Line. IEEE International Conference on Robotics and Automation 1999; 1:728–33.
[7] Peshkin M, Colgate E, Wannasuphoprasit W, Moort C, Akella P. Cobot Architecture, IEEE Transactions on Robotics and Automation 2001; 17 (4): 377–90.
[8] Krüger J, Bernhardt R, Surdilovic D, Spur G. Intelligent Assist Systems for Flexible Assembly. CIRP Annals - Manufacturing Technology 2006; 55 (1): 29–32.
[9] Krüger J, Lien T, Verl A. Cooperation of Human and Machines in Assembly Lines. CIRP Annals - Manufacturing Technology 2009; 58 (2): 628–46.
[10] Bannat A, Gast J, Rehrl T, Rösel W, Rigoll G, Wallhof F. Multimodal Human-Robot-Interaction Scenario: Working Together with an Industrial Robot. In Jacko JA, editor. Proceedings of the International Conference on Human-Computer Interaction. San Diego: Springer; 2009.
[11] Mayer M, Odenthal B, Faber M, Winkelholz C, Schlick C. Cognitive Engineering of Automated Assembly Processes. Human Factors and Ergonomics in Manufacturing & Service Industries 2014; 24 (3): 348–68.
[12] Kruse T, Pandey AK, Alami R, Kirsch A. Human-Aware Robot Navigation: A Survey. Robotics and Autonomous Systems 2013; 61 (12): 1726–43.
[13] Gombolay MC, Wilcox RJ, Diaz A. Towards Successful Coordination of Human and Robotic Work Using Automated Scheduling Tools: An Initial Pilot Study. Robotics: Science and Systems, Human-Robot Collaboration Workshop, Berlin, Germany 2013.
[14] Galindo C, Fernández-Madrigal JA, González J, Saffiotti A. Robot Task Planning Using Semantic Maps. Robotics and Autonomous Systems 2008; 56 (11): 955–66.
[15] Cucinotta T, Mancina A, Anastasi GF, Lipari G. A Real-Time Service-Oriented Architecture for Industrial Automation. Industrial Informatics IEEE Transactions Volume:5 Issue: 3 2009; p 267 – 277.
[16] Yinong C , Zhihui D , García-Acosta M. Robot as a Service in Cloud Computing. Fifth IEEE International Symposium 2010; 10.1109/SOSE.2010.44.
[17] Koubaa A. A Service-Oriented Architecture for Virtualizing Robots in Robot-as-a-Service Clouds. , Architecture of Computing Systems – ARCS, 27th International Conference, Lübeck, German 2014; pp 196-208.
[18] Yinong C, Xiaoying B. On Robotics Applications in Service-Oriented Architecture. Distributed Computing Systems Workshops 2008; pages 551-556.
[19] Diankov R, Kuffner J. OpenRAVE: A Planning Architecture for Autonomous Robotics. In Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213 2013.
[20] Koubaa A. A Service-Oriented Architecture for Virtualizing Robots in Robot-as-a-Service Clouds. , Architecture of Computing Systems – ARCS, 27th International Conference, Lübeck, German 2014; pp 196-208.
[21] Guoqiang H, Wee Peng T, Yonggang W. Cloud robotics: architecture, challenges and applications, Network. IEEE , 26:3, p 21-28 2012.
[22] Makris S, Tsarouchi P, Surdilovic D, Krüger J. Intuitive dual arm robot programming for assembly operations, CIRP Annals - Manufacturing Technology, 2014; 63/1:13–16.
[23] Process Simulate for Robotics and Automation, (2015), http://www.plm.automation.siemens.com/en_us/products/tecnomatix/manufacturing-simulation/human-ergonomics/process_simulate.shtml
[24] Safe camera system SafetyEYE, (2015), https://www.pilz.com/eshop/b2b/catalogstart/(layout=7_67_4_66_6&uiarea=0)/.do
[25] Tsarouchi P, Makris S, Michalos G, Stefos M, Fourtakas K.,Kaltsoukalas K, Kontovrakis D,Chryssolouris C. Robotized Assembly Process Using Dual Arm Robot, Procedia CIRP, 2014;23:47–52.
[26] Robot Operating System (ROS), 2015, http://www.ros.org/.
[27] ROS Groovy Galapagos, 2015, http://wiki.ros.org/groovy.

ID	Title	Pages
1699250	ROS Based Coordination of Human Robot Cooperative Assembly Tasks-An Industrial Case Study ☆	6