Concept and Architecture for Programming Industrial Robots using Augmented Reality with Mobile Devices like Microsoft HoloLens

Jan Guhl and Son Tung Nguyen
Department of Industrial Automation Technology
Institute for Machine Tools and Factory Management
Technische Universität Berlin, Germany
Email: guhl@iat.tu-berlin.de

Jörg Krüger
Fraunhofer Institute for Production Systems
and Design Technology
Berlin, Germany

***Abstract*—This paper proposes a concept for human-robot interaction using techniques of virtual and augmented reality on mobile devices like cell phones and tablets or mixed reality devices like the HoloLens. By combining data received from real robots together with the perception and abilities of a human operator innovative applications are imaginable. Visualizing not only the current robot state but also the robots environment captured with different sensors and processed with both machine and human vision can lead to a rising percentage of robot assisted workplaces or robot installations. Since the visualization of and the interaction with the robotic application is not locally restricted new or improved use cases like remote maintenance or faster startup of industrial robots can be realized. Therefore, an architecture split into three functional units and an overview of our implementation is presented.**

Human-robot interaction, Robot programming, Augmented reality, Virtual reality, Mobile Devices, HoloLens

I. Introduction

Modern production and manufacturing facilities and processes are challenged with a decreasing batch size down to size one due to increasing individualization of goods and products. In times of Connected Industry and Internet of Things [1] this means a rising demand for flexible and reconfigurable production systems. A flexible chain of operational tools can therefore provide potential for saving costs. Another challenge directly linked to flexible production lines is their often needed reconfiguring. Concerning industrial robots this means reprogramming of their trajectories and paths. The task of programming industrial robots nowadays mostly concentrates on writing vendor and device specific source code which contains movement and logical instructions. This means that the resulting programs are not portable from one machine to another and therefore every even slight change in the production line can lead to a high effort in terms of both time and costs.

Every robot and robotic system can be seen as a cyber-physical system according to the Reference Architecture Model for Industrie 4.0 [2]. Following the idea of Industrie 4.0 and the Connected Industry, the field device can be separated from the control device. These control devices can then be instantiated as a hardware independent cloud-service running on for example a virtual server in a cloud infrastructure. By separating the machine and its control device from another new problems arise.

One challenge is to monitor and supervise the exchanged data between different devices in such a cloud based control infrastructure. When robotic systems are controlled from a control device running as a virtual server it is not simply possible to visualize the motions that are going to be performed by the industrial robot. In this paper we present an approach to this problem.

II. Related Work

A. Human-Robot Interaction

Human-robot interaction is a broad field of research. Every application dealing with either the simplification of robot programming or only robot programming itself can be taken into account. Human-Robot Interaction has many important usages in modern life and collaboration between robots and humans can be beneficial in many aspects, e.g. health care or manufacturing contexts. In [3] the authors describe a multi-modal human-robot interface by mimicking the way people communicate with each other. Focusing on the modality of natural language and gestures. An user input will be processed to directly trigger an action by mobile robots. It is stated that an operator is less concerned with how to interact with a robotic system but can concentrate on the tasks and goals.

A case study on remote programming of robotic manipulators is presented in [4]. The paper focuses on the remote programming of robots, called the lowest level of interaction between the operator and the robot. An interface to a telelaboratory is described which allows users to connect and program several features, like cameras and the robot control. Included are different visualizing methods like augmented and virtual reality. Augmented reality is described as a technique to enhance the way operators interact with robotic scenarios, since receiving and publishing information from and into the system is simplified.

When it comes to physical human-robot interaction one challenge is how the robot and the human are placed spatially to another since the robot has to be able to avoid collisions with the human but at the same time be able to share its workspace with the human in order to enable human robot collaboration. An approach to automatically determine the Interaction Workspace as a representation of accessible space by multiple agents is presented in [5].

A different aspect of human-robot interaction is explored in [6]. The authors examine the influence of the operators age on the acceptance and basic understanding of robot behaviors. Different aspects of the interaction are investigated with a series of video-based tests. The results show similarities between teenagers and adults and that a robot behavior designed for adults would be probably also effective in the interaction with teenagers.

The approaches are all closely related in terms of using visual modality for enabling human-robot interaction showing thereby the importance of human vision for a natural interaction with robotic systems.

B. Distributed Machine Communication

Every application where some needed data and or code is obtained or processed in the cloud can be taken into account. Some approaches use established solutions and middleware like the Robot Operating System (ROS) [7] for establishing the communication in Cloud-Based robot control. Rapyuta [8], a Cloud-Robotics platform, is tailored towards multiprocess high-bandwith robotics applications and allows the outsourcing of computation tasks but is not focused on realizing real-time requirements. Based on [8], the authors in [9] built a system which provides Path-Planning-as-a-Service.

Other approaches use a custom tailored communication layer. In [10] Niebuhr et al. develop a concept for a network service framework which creates a distributed communication graph between systems. The framework is realized using existing components like ZeroMQ[1] for message passing and Protocol Buffers[2] for data serialization and deserialization.

Trying to realize an Automation-as-a-Service infrastructure is the goal of [11]. Vick et al. show an implementation of a distributed control system using a model predictive feedback control system to compensate communication delays in networked industrial robot control.

III. Concept and Implementation

As seen in the related work a key concept for human-robot interaction is that the human is able to foresee and predict the robots actions and movements. Therefore, a system which makes use of the human perception to enable a natural interaction is proposed. As modern robotics tend to use distributed control systems and algorithms the infrastructure shall be easily adoptable and integrable to those cloud based approaches.

[1]ZeroMQ - http://zeromq.org/

[2]Protocol Buffers - https://developers.google.com/protocol-buffers/

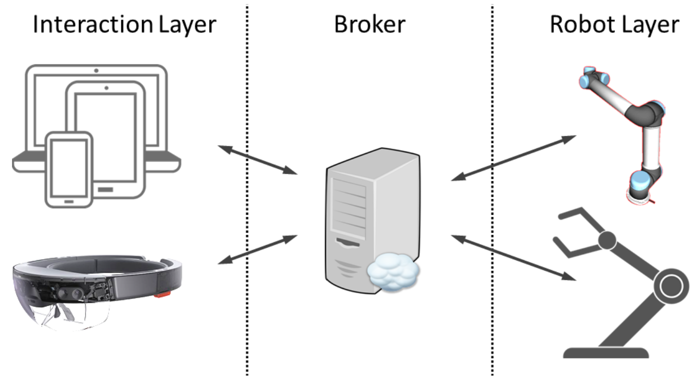

Fig. 1. The realized architecture. The Interaction Layer represents all mobile devices which communicate via the broker with all involved robots on the Robot Layer

The goal of this work is to develop an architecture that allows interaction with robotic system, especially industrial robots. Which means the visualization of robot programs and any other useful data associated with industrial robots, for example the workspace of the robot or movement restrictions as safety planes or joint angle limits as well as interacting with the robot which means manipulating the robot for example by changing joint angles or create programs for the targeted robot.

In figure 1 the implemented architecture is shown. We propose to divide the framework into simple layers according to their functions. First there is the Robot Layer, containing all robotic system that shall be interacted with. In the middle there is the Broker, which is aware of every connected device, both robots and interaction devices. In our case the robots are

- Universal Robots UR 5,
- Comau NJ 130 - 2.6 and
- KUKA KR 6.

Those robots are very different from another. While the first one is a lightweight manipulator specifically designed for human-robot collaboration, the others are classic industrial robots that can realize high velocities and accelerations, potentially harmful to humans. Which is why it is beneficial to be aware of the robots movements before executing an already existing program.

A. Architecture

The robot layer is responsible for connecting all robots to the Broker. Therefore, it needs to communicate to a variety of different robot controls with varying interfaces. The challenge is to find a generalized robot description that can be sent and understood by the broker. This description format has to include all valuable information that the robots can offer. Starting by the count of axes and the spatial dimensions of the manipulator but also including information about for example available tools or force control abilities and its actuator gains. As of now we concentrate on the movements of the robots and use axes angles to represent a robot state. To join the robots to the broker each one needs its own connector implementation.

(a) (b)

Fig. 2. Visualizing a model of an UR5 robot captured on a HoloLens using our system. (a) The robot is placed relative to a marker to ensure correct position and orientation from different viewpoints. (b) Virtual model side by side to a real robot to compare the robots dimensions.

The broker is the central instance in the architecture. This instance is implemented as a server for both the robots and the interaction devices as they have to actively connect themselves. In this manner it is ensured that the devices do not have to know anything on beforehand about each other. Since the whole conversation is done via the broker it can be ensured that each device only has to handle a single connection to the broker but still is able to receive information and data from every other connected machine.

The interaction layer is responsible for presenting all available information to the end user and enable the human-robot interaction. In this work we focus on mobile devices for interfering with the machines since we believe that many sensory modalities can be implemented using these types of devices and therefore a natural form of interaction can be realized in the future. Light modality or the visual part can be obtained with a simple display. We chose augmented reality in order to enrich the users environment with additional information. Sound modality can be stimulated with built-in speakers or headphones. Even the tactile perception can be triggered by using vibration feedback.

In figure 2a a screenshot from the running interaction application can be seen. There, a model of the UR 5 is visualized using augmented reality techniques on a marker target which is simply an image of the robot itself. A visualization of the robot and the original robot can be seen in figure 2b. Using the virtual robot alongside the real one enables the comparison of the spatial dimensions of the robots.

B. Implementation

Each layer of the architecture has different requirements and is executed on very different hardware. The fundamental request on the robot layer is to connect different robots of varying vendor and type to the broker. Therefore, its implementation has to be robot specific but expose the same interface to the broker. In case of the UR5 the interface is provided via an ethernet protocol. We used the implementation from Andersen [12] to connect the robot to the broker. The interface to the KUKA KR6 is a serial port. On the robot control a program that interprets commands sent over the interface is running which translates the movement instructions into the robot specific language KUKA Robot Language (KRL).

The broker is the central part in this concept. Since every communication is at least initially managed by it, implementing the broker demands a high performance and scalability for handling numerous connections and devices. Therefore we develop this part of software in C++.

The interaction layer has to be platform independent and include as many devices as possible. Mobile devices are quite different from another looking closely at the technical specifications. For example cell phones and tablets are targeting different user groups and therefore vary a lot in terms of hardware performance, operating systems and user interface. Tested devices include

- 9.7 inch Android Tablet,
- 13.5 inch Windows Tablet,
- Microsoft HoloLens and
- Laptop running Windows 10.

For rendering the models and visualizing the user interface we chose to use a game engine, namely Unity[3] as it was the only engine supporting the HoloLens at time of choosing. A robot is represented by a so called Prefab, similar to templates known from other languages. To enable spatial positioning of the virtual robot models we utilize marker tracking as a base for our augmented reality. The used software is Vuforia[4].

The layers and devices communicate with each other using TCP/IP since it is supported by nearly every modern mobile device for example via Wireless LAN. The protocol wrapping

[3]https://unity3d.com/
[4]https://www.vuforia.com/

the transmitted data is custom tailored to our needs and involves several steps:

1) Establish connection to the broker.
2) Request list of connected devices, both from the interaction and the robot layer.
3) Receive general information about the requested devices e.g. axes angle limits, maximum payloads or possible types of interaction.
4) Request a set of detailed information about some devices with precision.
5) Receive this set of information with a high frequency.
6) Disconnect from the broker on program exit.

The data exchanged between the different layers is serialized and deserialized using Protocol Buffers. Which offer interfaces to many programming languages like C++ used in the broker or C# used inside of Unity.

IV. Conclusion and Future Work

We presented a concept and architecture for using mobile devices like the Microsoft HoloLens for interacting and programming robotic systems particularly industrial robots. The robots and mobile devices are connected to each other via a so called broker and communicate using Protocol Buffers on top of TCP/IP connections. Using established software components like a game engine enables the support for a variety of mobile devices running on different operating systems and hardware performance levels.

As of now, transmitting axes angles from and to a robot is supported. Since robot programs are a set of axes angles given for specific timestamps the possibilities arise. Existing robot programs can be visualized and evaluated on the real robot in its current setup before execution which enables an operator to decide if for example a change in the robot cell requires a change to the current robot program.

Future work will be focusing on increasing the possible types of interaction like manipulating existing robot programs or integrating features from established middleware like path planning from ROS. Including not only virtual objects but also the virtual representation of objects and components of the robots environment obtained by different sensors and processed by both machine and human vision enables a new form of human-robot interaction. Using our method innovative human-robot interaction scenarios are imaginable. By combining data received from a real robot with the perception and abilities of a human operator can be beneficial for example in cases of disaster management, remote maintenance or education facilities.

References

[1] Ashton, K., That 'Internet of Things' Thing, 2009, available at http://www.rfidjournal.com/articles/view?4986 (18.05.2017)

[2] DIN 91345: 2016-04, Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)

[3] D. Perzanowski, A. C. Schultz, W. Adams, E. Marsh and M. Bugajska, "Building a multimodal human-robot interface", in IEEE Intelligent Systems, vol. 16, no. 1, pp. 16-21, Jan-Feb 2001.

[4] R. Marin, P. J. Sanz, P. Nebot and R. Wirz, "A multimodal interface to control a robot arm via the web: a case study on remote programming", in IEEE Transactions on Industrial Electronics, vol. 52, no. 6, pp. 1506-1520, Dec. 2005.

[5] N. Vahrenkamp et al., "Workspace analysis for planning human-robot interaction tasks", 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 1298-1303.

[6] F. Rea, P. Muratore and A. Sciutti, "13-year-olds approach human-robot interaction like adults", 2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), Cergy-Pontoise, 2016, pp. 138-143.

[7] M. Quigley et al., "ROS: an open-source Robot Operating System", 2009 Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA), 2009

[8] G. Mohanarajah, D. Hunziker, R. D'Andrea and M. Waibel, "Rapyuta: A Cloud Robotics Platform", in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2, pp. 481-493, April 2015.

[9] M. L. Lam and K. Y. Lam, "Path planning as a service PPaaS: Cloud-based robotic path planning", Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on, Bali, 2014, pp. 1839-1844.

[10] C. A. Niebuhr and A. Vick, "Distributed real-time control service framework for human-robot interaction applications," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, 2016, pp. 1-8.

[11] A. Vick, J. Guhl and J. Krüger, "Model predictive control as a service - Concept and architecture for use in cloud-based robot control", 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, 2016, pp. 607-612.

[12] T. T. Andersen, "Optimizing the Universal Robots ROS driver.", available at http://orbit.dtu.dk/en/publications/optimizing-the-universal-robots-ros-driver(20dde139-7e87-4552-8658-dbf2cdaab24b).html, (18.05.2017) 2015