Multimodal binding of parameters for task-based robot programming based on semantic descriptions of modalities and parameter types

Alexander Perzylo[1]* Nikhil Somani[1]* and Stefan Profanter[1] and Markus Rickert[1] and Alois Knoll[2]

***Abstract*— In this paper, we describe our ongoing efforts to design a cognition-enabled industrial robotic workcell, which significantly increases the efficiency of teaching and adapting robot tasks. We have designed a formalism to match task parameter and input modality types, in order to infer suitable means for binding values to those parameters. All modalities are integrated through a graphical user interface, which a human operator can use to program industrial robots in an intuitive way by arbitrarily choosing modalities according to his or her preference.**

I. Introduction

Programming industrial robots can be a tedious task. Typically, a lot of expert knowledge in the domain of robotics is required to implement even simple actions. Human operators require weeks of training and decision makers might be afraid of rendering their company dependent on just a few workers capable of using their robots. The alternative of hiring a system integrator is a valid choice only if the corresponding extra costs can be covered through longer product life cycles.

As a result, the assessment of financial viability of deploying robot-based automation solutions is highly influenced by the ratio of programming time of the robot and number of produced goods. The rate of adoption of such robot systems for small and medium-sized enterprises (SMEs) is burdened as SMEs often deal with small lot sizes or even individualized products.

In order to overcome such limitations, novel robot teach-in paradigms must be developed to allow non-experts in the domain of robotics to efficiently program robots to cope with new tasks.

In the field of service robotics, the focus is on the development of completely autonomous robots, that follow a goal-based programming approach. The level of detail required for instructing such robots is very low. The user consequently also has very little control over the actual execution of the robot program.

With respect to these extremes in robot programming paradigms, our approach tries to integrate the best features from both approaches (Fig. 2).

We propose a natural teaching paradigm that aims at establishing a high-level communication layer between human operator and robot system. This layer can be maintained through the modeling of knowledge about industrial domains, processes, workpieces, and workcells in a semantic and machine-understandable way. Having prior domain-relevant knowledge available on both sides, instructions can be abstract and not all process parameters have to be explicitly specified by the operator. Some of the missing parameters can be automatically inferred.

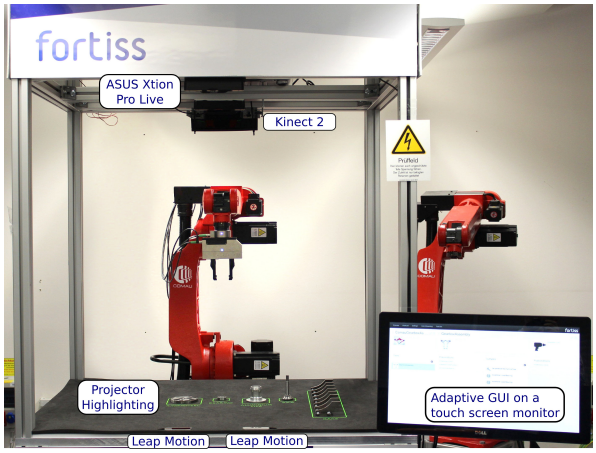


Fig. 1: Cognitive robotic workcell with intuitive graphical user interface integrating further input and output modalities

One aspect of naturally teaching tasks to a robot system is an adequate selection of communication modalities. Apart from the default teach pendant, many input modalities for robot systems have been researched, e.g., natural language in spoken or written form, hand or body gestures, pointing devices, and augmented reality interfaces.

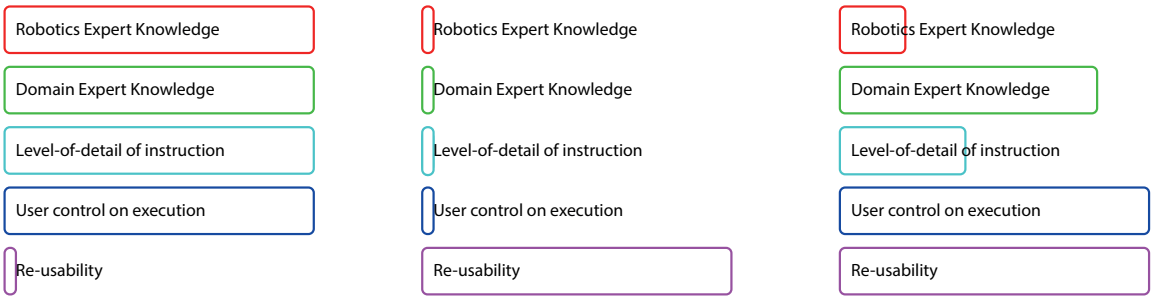
As an extension to our approach to build an easy-to-teach cognitive robotic workcell (Fig. 1), we describe available communication modalities as part of a semantic workcell description. By combining this information with a semantic process description, the robot system can infer compatible modalities for setting specific types of process parameters. An intuitive touch-enabled graphical user interface acts as the central teaching component assisting the operator in using different modalities.

In this work, we present a robot programming interface targeted towards efficient and intuitive teaching of industrial robots. Designing one interface that is optimal for all scenarios and robot tasks is a difficult and probably infeasible task. Instead, we use a multimodal approach where the user can switch seamlessly between different modalities.

[1]Alexander Perzylo, Nikhil Somani, Stefan Profanter, and Markus Rickert are with fortiss GmbH, Guerickestr. 25, 80805 München, Germany {perzylo|somani|profanter|rickert}@fortiss.org

[2]Alois Knoll is with the Department of Informatics VI, Technische Universiät München, Boltzmannstr. 3, 85748 Garching, Germany knoll@in.tum.de

* These authors contributed equally to this work.



(a) Classical teaching on teach pendant based on robot programming language

(b) Instruction of fully autonomous service robot

(c) Our approach of intuitive teaching based on semantic process descriptions

Fig. 2: Illustrative comparison of different robot teaching paradigms with respect to required knowledge, re-usability and level-of-detail of required instructions. This diagram is not meant to be interpreted as a formal evaluation.

II. Related Work

Using multimodal input technologies for interacting with a system has various advantages resulting in a more flexible and reliable system [1], [2]. An approach for programming industrial robots using markerless gesture recognition is evaluated in [3]. The authors conclude that using a multimodal system results in a significant reduction of required teach-in time.

[4] evaluate different gestures (e.g., point at part, twist, swap, cover) for industrial use-cases and create a gesture lexicon to map those gestures to a semantic meaning.

III. Testbed: cognitive robotic workcell

Fig. 1 shows the robotic workcell used in our system. The workcell features several sensors that enable multimodal input. The calibration and data synchronization of input streams from these sensors is done using utilities provided by the Robot Operating System (ROS). The involved devices and their placement is described in the following sections.

1) Microsoft Kinect 2: A newly released motion sensing RGBD sensor, with better accuracy, wider field of view and higher image resolution then its predecessor, is placed on the top of the workcell, frontally facing the human worker. The sensor and the corresponding software (Kinect for Windows SDK 2.0) is capable of tracking 25 skeleton joints. This articulated human skeleton tracking information is used for detecting body gestures and human activities [5].

2) ASUS Xtion Pro Live: This RGBD sensor provides registered point clouds. It is placed on top of the metal cage facing towards the tabletop. The RGBD data obtained from this sensor is used for detecting objects on the table [6]. The device is similar to the first generation of the Microsoft Kinect sensor, but it is more compact, lighter and does not require an external power supply.

3) Leap Motion Sensor: The Leap Motion sensor is used to track the complete articulated hand with sub-millimeter accuracy [7]. Due to its small size, accurate hand tracking can only be achieved within a very limited area around the sensor (in a distance of 25mm to 600mm). Hence, two Leap Motion sensors were integrated inside the table surface directly facing the expected working area of one hand.

4) Projector: A projector is used to provide visual feedback on the tabletop. We use a DLP projector having a brightness of 6500 lumen, which is necessary for obtaining a sharp image under well-lit conditions. A first surface mirror is used to redirect the projection of the horizontally mounted projector and to cover the entire tabletop (of size 120cm x 90cm) from a short vertical distance.

IV. Multimodal Semantics

This approach is based on semantic descriptions of robot processes [8], workcells, and deep object models [9].

In our representation, robot processes consist of graphs of hierarchically defined tasks. Each of these tasks has a set of required and optional parameters. Workcell models specify the structural entities of a workcell, i.e., robots, tables, sensors, tools, workpieces, and abstract capabilities that describe the available skills of the system. Available modalities can be derived from the workcell model by exploiting information about the sensors and software capabilities. Apart from a common meta-description, our object models also link to deep representations of the objects' geometries, which are based on a boundary representation [9]. This allows us to select objects using our multimodal interface and to specify geometric constraints between subparts of their geometries, using the same data model. This is particularly useful for the constraint-based definition and execution of assembly tasks [10].

Semantic descriptions of these modalities and parameter types (Fig. 3) enable an adaptive user interface that can filter modalities based on the current workcell, task domain, as well as user preferences. The GUI then presents the options for choosing suitable modalities that can be used to specify a parameter of a particular type (Fig. 4). The modalities used in our system were selected based on the results of a user study on modality preferences in industrial robotic workcells [11].

V. Multimodal User-Interface

Our robot programming approach is object centric, i.e., tasks and their parameters are defined in terms of semantically described entities such as workcells and workpieces. An object (e.g., a workpiece) to be used as a task parameter

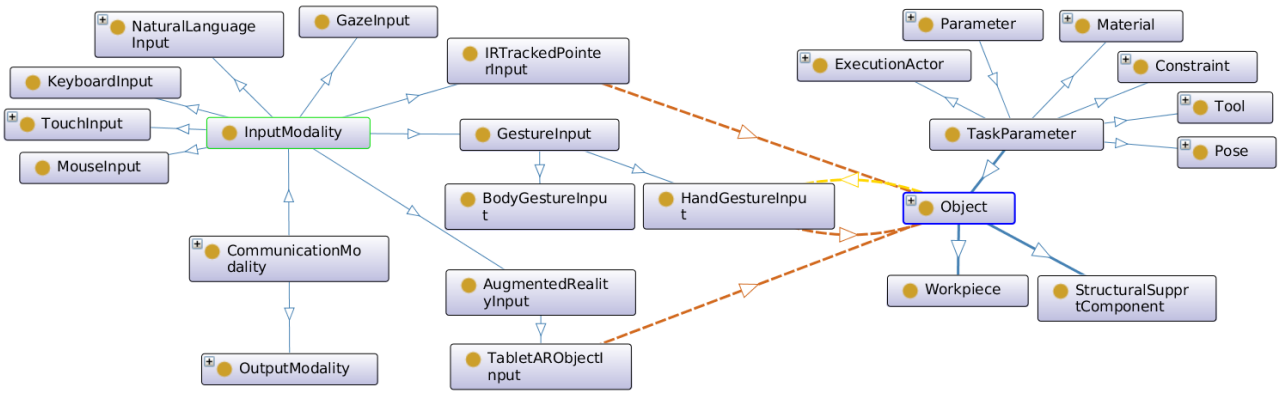
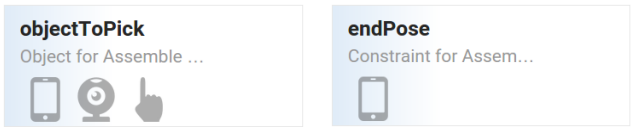


Fig. 3: Taxonomy of input modalities and task parameter types (blue arrows represent subclass relations). This excerpt shows that the input modalities *IRTrackedPointInput*, *HandGestureInput* and *TabletARObjectInput* provide values of parameter type *Object* (orange dashed arrows). The *preferredModality* (yellow dashed arrow) for this parameter type is set to be *HandGestureInput*.

can be selected using multiple modalities. Our multimodal approach filters the available modalities based on workcells, task descriptions and parameter types (Fig. 4).



(a) Parameter *objectToPick* is of type *Object* and can be set by different modalities, i.e., *touch input*, *augmented reality selection* and *pointing gesture*

(b) For setting parameter *endPose*, which is of type *GeometricInterrelationConstraint*, only the modality *touch input* is available

Fig. 4: Example of filtered modalities per parameter type

The supported modalities are briefly described in the following subsections.

A. Touch input

Using the graphical user interface, a list of object models with thumbnails is presented to the user. The display supports touch input which allows the user to intuitively interact with the system. This list is filtered based on the selected application domain and the objects available in the workcell (Fig. 5).

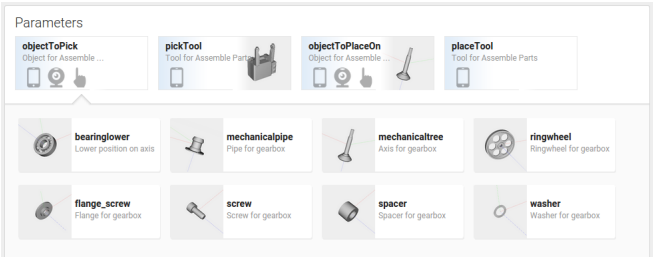


Fig. 5: Filtered list of object models available for selection in the graphical user interface

B. Pointing at objects

A projector is used to highlight the detected objects in the workcell and to project additional meta-data below them (e.g., name, dimensions, weight). The Asus Xtion Pro camera, mounted in a table-top configuration (see Section III), provides RGBD data which is used for CAD-based object detection [6]. Two Leap Motion sensors, mounted on the tabletop (see Section III) provide articulated hand poses. By combining the object positions and pointing direction, the system can determine the object to be selected (Fig. 6).

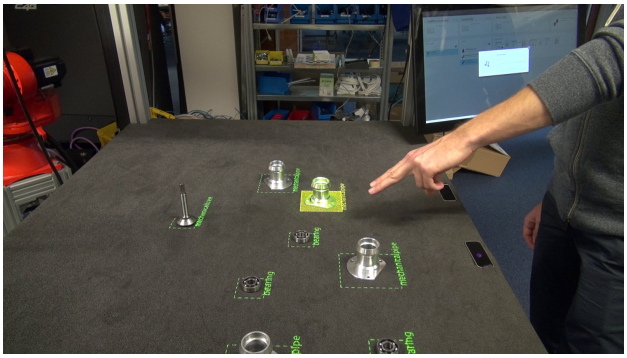


Fig. 6: Detected and highlighted objects on the tabletop. The human operator selects an object by pointing to it. The selected object is highlighted on the tabletop and the GUI

C. Selection using AR on a tablet

This interface is available on tablets, where objects are recognized and highlighted in the image provided by the tablet camera. The user interface shows the augmented camera image, where the objects are detected using the 2D image. The user can click on the detected objects and select them (Fig. 7).

VI. Evaluation of Modality Preferences

In our previous work, we conducted a user study to analyze and model modality preferences in industrial human-robot interaction scenarios [11]. The study was built up

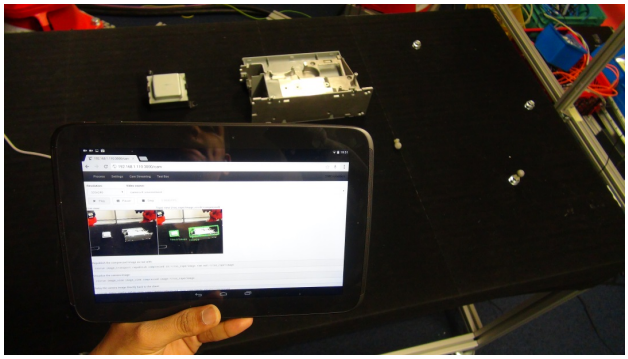


Fig. 7: Selecting an object from a tablet camera image augmented with detected objects

as a Wizard-Of-Oz experiment using the cognitive robotic workcell described in Section III. The goal of this study was to evaluate which input modality (among touch, gesture, speech, and pen-like pointing device) is preferred by the user for each parameter type. Finally, the results were modeled using a semantic description language which could then be used in our workcell to make the interaction and teach-in process easier and more intuitive.

30 participants were included in the evaluation where the majority of the subjects had technical background knowledge (especially in the field of robotics and embedded systems). The average age of the participants was 27 years.

Each participant had to perform different programming tasks using all four input modalities sequentially. These programming tasks covered different domains: assembly, pick-and-place, and welding. After the practical part, the user was asked to fill out a set of questionnaires to state their impressions about the system and preferred input modalities for specific tasks.

The evaluation of these questionnaires shows that there are some significant differences between the most and least preferred modality, confirming our hypothesis (Fig. 8). Gesture input was selected as the most preferred input modality (p-Value < 0.0001), while touch input and 3D pen input were nearly equally rated second in order of preference. Speech input was by far the least preferred modality (p-Value < 0.0001).

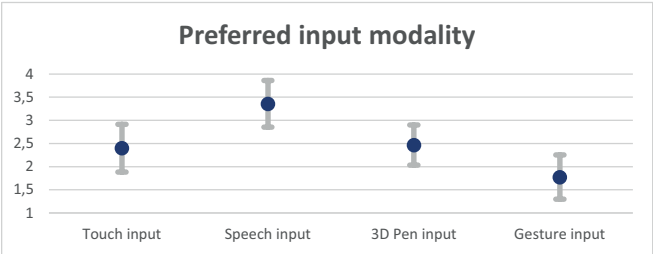


Fig. 8: Evaluation of preferred input modalities. Gesture input is most preferred, speech input least preferred. The *blue* dot marks the average mean over all participants, the *gray* bar represents the standard deviation.

VII. Conclusions and Future Work

In this work, we presented a multimodal interface for human-robot interaction, specifically targeted towards the teaching of robot processes. The system is designed to be intuitive and efficient, as well as flexible towards switching between I/O modalities or even inclusion of new modalities in the system. The multimodal system was demonstrated using three different modalities for selecting an object as a task parameter. This can be extended in the future to handle more parameter types and include more modalities based on enhanced capabilities of newer sensors. In the current system, the human-robot interaction is limited to the teach-in phase. The multimodal system could be extended to enable user interaction during robot task execution.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287787 in the project SMErobotics.

References

[1] P. Cohen, M. Johnston, D. McGee, and S. Oviatt, "The efficiency of multimodal interaction: a case study," in *International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.

[2] S. Oviatt, R. Lunsford, and R. Coulston, "Individual Differences in Multimodal Integration Patterns : What Are They and Why Do They Exist?" in *Conference on Human Factors in Computing Systems (CHI)*, New York, USA, 2005.

[3] J. Lambrecht and J. Krüger, "Spatial Programming for Industrial Robots: Efficient, Effective and User-Optimised through Natural Communication and Augmented Reality," *Advanced Materials Research*, vol. 1018, pp. 39–46, Sept. 2014.

[4] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry: Intuitive human-robot communication from human observation," in *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, ser. HRI '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 349–356.

[5] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll, "Human activity recognition in the context of industrial human-robot interaction," in *AsiaPacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Siem Reap, Cambodia, December 2014.

[6] N. Somani, E. Dean-Leon, C. Cai, and A. Knoll, "Scene Perception and Recognition in industrial environments," in *9th International Symposium on Visual Computing (ISVC'13)*. Springer, July 2013.

[7] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[8] A. Perzylo, N. Somani, S. Profanter, M. Rickert, and A. Knoll, "Toward efficient robot teach-in and semantic process descriptions for small lot sizes," in *Proceedings of Robotics: Science and Systems (RSS), Workshop on Combining AI Reasoning and Cognitive Science with Robotics*, Rome, Italy, July 2015, http://youtu.be/B1Qu8Mt3WtQ.

[9] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *IEEE/RS International Conference on Intelligent Robots and Systems (IROS)*, September 2015.

[10] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with CAD semantics: from intuitive specification to real-time control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015.

[11] S. Profanter, A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "Analysis and semantic modeling of modality preferences in industrial human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015.