Elektrotechnik & Informationstechnik (2019) 136/7: 326–333. https://doi.org/10.1007/s00502-019-00741-4

# Skill-based programming of complex robotic assembly tasks for industrial application

S. C. Akkaladevi, A. Pichler, M. Plasch, M. Ikeda, M. Hofmann

In recent years, a paradigm shift is underway as robots leave their typical application field and move into domains that have been untouched by robotic automation. These new kinds of automation systems allow more product variations, smaller life cycles, smaller batch sizes and pave the way from mass production to mass customization. This is due to completely new breed of safe robot technology but also novel ways of setting up new applications like e.g. kinesthetic programming. However, the topic of reducing the programming effort for complex tasks using natural modes of communication is still open. This paper addresses the key developments in this field, shows different ways of programming, and gives relevant use cases in industrial assembly. The technology coverage starts with an online workflow editor called XROB that allows easy-to-use setup of process workflows and related skill parameters. However, in order to reduce the programming effort, a novel way to demonstrate process trajectories by using instrumented hand guided process tools is presented. Finally, the paper gives an overview of a promising approach that allows programming without touching the robot just by demonstrating the process by an expert. The semantic relations between activities executed by the human and robot skills are captured to learn the task sequence of the assembly process. The acquired process knowledge is refined to execute robotic tasks with the help of an interactive graphical user interface (GUI). The system queries the user for feedback, asking for specific information to help the robot complete the task at hand. The given examples show the usability of flexible programming tools in the automation chain and the presented results provide strong evidence of the technological potential in the field.

Keywords: skill-based learning; human-robot cooperation

***Skill-basierte Programmierung von komplexen Roboter-Montageaufgaben für die industrielle Applikation.***

*Die produzierende Industrie erfährt einen Paradigmenwechsel, da Roboter ihr typisches Anwendungsfeld verlassen und in Bereiche vordringen, die von der robotergestützten Automatisierung bisher unberührt geblieben sind. Diese neuartigen Automatisierungssysteme ermöglichen mehr Produktvariationen, kleinere Lebenszyklen und kleinere Losgrößen und ebnen den Weg von der Massenproduktion zur Produktindividualität. Dies ist auf eine völlig neue Art sicherer Robotertechnologie zurückzuführen, aber auch auf neue Wege zur Realisierung neuer Anwendungen wie z.B. der kinästhetischen Programmierung. Ein weiterhin offenes Thema ist den Programmieraufwand für komplexe Aufgaben mit natürlichen Kommunikationswegen zu reduzieren. Diese Publikation befasst sich mit den wichtigsten Entwicklungen in diesem Bereich, zeigt verschiedene Möglichkeiten der Programmierung auf und beschreibt relevante Anwendungsfälle in der industriellen Montage. Das betrachtete Technologieportfolio beinhaltet einen Online-Workflow-Editor namens XRob, welcher die einfache Einrichtung von Prozessabläufen und damit verbundenen Qualifikationsparametern ermöglicht. Um den Programmieraufwand zu reduzieren, wird eine neuartige Möglichkeit zur Demonstration von Prozessverläufen, unter Verwendung von instrumentierten, handgeführten Prozesswerkzeugen, vorgestellt. Schließlich gibt das Paper einen Überblick über einen vielversprechenden Ansatz, der die Programmierung ohne Berührung des Roboters ermöglicht, nur durch die Demonstration des Prozesses durch einen Experten. Die semantische Beziehung zwischen den vom Menschen ausgeführten Tätigkeiten wird erfasst, um die Aufgabenreihenfolge des Montageprozesses zu erlernen. Das erworbene Prozesswissen wird mit Hilfe der interaktiven grafischen Benutzeroberfläche (GUI) zu Roboteraufgaben verfeinert. Das System fragt den Benutzer nach Feedback und fragt nach spezifischen Informationen, die dem Roboter helfen, die anstehende Aufgabe zu erfüllen. Die angeführten Beispiele belegen die Benutzerfreundlichkeit flexibler Programmierwerkzeuge in der Automatisierungskette, und die vorgestellten Ergebnisse zeigen das technologische Potenzial in diesem Bereich auf.*

*Schlüsselwörter: Skill-basiertes Lernen; Mensch-Roboter Kooperation*

Received July 29, 2019, accepted September 26, 2019, published online October 31, 2019

## 1. Introduction

As manufacturing demographics change, advances in human-robot interaction in industries have taken many forms. However, the topic of reducing the programming effort required using natural modes of communication is still open. In the literature, different methods for developing robotic agents, which can learn from a human instructor, are described. Common to all methods is the reduction of programming effort. The popular approaches in literature include Learning by Demonstration (LbD) [3], Learning by Programming (LbP) [1] and Learning by Interaction (LbI) [4]. LbD involves a physical demonstration of the process by the human operator. The system captures process relevant features and maps them to

**Akkaladevi, Sharath Chandra,** Profactor GmbH, Im Stadtgut A2, Steyr-Gleink 4407, Austria (E-mail: sharath.akkaladevi@profactor.at); **Pichler, Andreas,** Profactor GmbH, Im Stadtgut A2, Steyr-Gleink 4407, Austria; **Plasch, Matthias,** Profactor GmbH, Im Stadtgut A2, Steyr-Gleink 4407, Austria; **Ikeda, Markus,** Profactor GmbH, Im Stadtgut A2, Steyr-Gleink 4407, Austria; **Hofmann, Michael,** Profactor GmbH, Im Stadtgut A2, Steyr-Gleink 4407, Austria

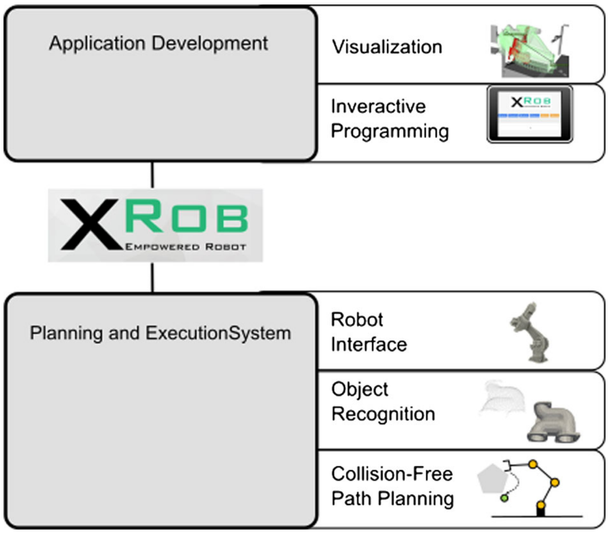


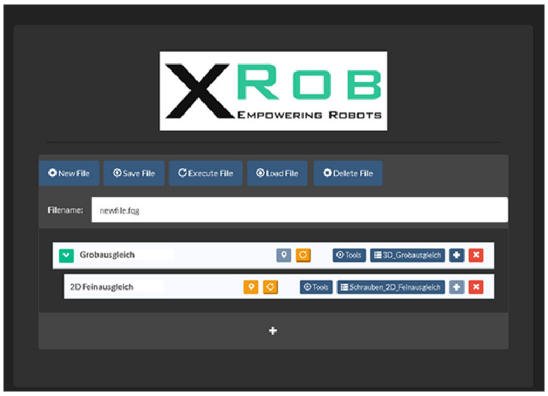
**Fig. 1. Overview on the software components within the XRob framework**

the robot's embodiment. The robot then aims to learn and execute the complete process with the help of extracted features. However, depending on the process complexity it could be quite challenging to extract the features at the required granularity for the robot to learn all the relevant features. LbP on the other hand uses Task Frame Formalism (TFF). TFF deals with lower level entities, usually called robot skills which instantiate actions. From the perspective of the robot operation, any given assembly task is (and could be) broken down into a form of skills which the robot can interpret. The main drawback of this approach is that the modeling complexity increases exponentially as the complexity of the task increases. In LbI, the agent learning process (using for e.g. Reinforcement Learning) takes place over discrete time steps by interacting with the environment and gaining experience about the outcome [4]. However, to reach an optimal policy (the set of actions that lead to the maximum reward) a substantial interaction with the environment is required.

Each of the above approaches (LbD, LbP, LbI) has its own advantage as well as corresponding drawback. This work aims to combine the above approaches to teach a goal oriented assembly process to the robotic system. First, a TFF based framework (XRob) for easily editing work-flows is described in Sect. 2. Section 3 presents a LbD approach using instrumented tools. The framework that exploits LbD to learn the coarse knowledge of the assembly process and then later refine the knowledge using LbP and LbI approaches is described in Sect. 4. And finally some concluding remarks are given in Sect. 5.

## 2. The XRob framework

The XRob software framework [2] enables the creation of complex robot applications within fewer minutes. It builds on unique, easy-to-use features that significantly speed up commissioning and make the operation more cost-efficient and flexible than common programming methods. The special software architecture allows easy and intuitive creation of processes and configuration of the components of a robot system via a single user interface. Figure 1 provides an overview on the software components within the XRob framework. The framework exploits the Learning by Programming (LbP) paradigm [1].



Filename: newfile.fqg

Grobausgleich

2D Feinausgleich

**Fig. 2. Online programming interface for the XRob framework**

The main strength of the framework is the strong abstraction for the end user, while keeping the customization capabilities of the system. This means, that the end user can focus on the actual process to be executed by the robot. This is achieved by reducing the input needed by the user to process position and the admission to the robot to collect additional information of the information's. The second input includes the permission to move the robot. The user interface which is shown to the user is shown in Fig. 2.

At the same time, the system can be customization by an engineer to fit the environment. One important adaption to the environment is the collision scene and the used objects. These information are used to navigate the robot in the environment without any collision. Also deviations of manipulated parts are compensated with 3D object localization algorithms.

**Object Recognition:** The aim of 3D Object Recognition is to localize the pose and position of an object of interest in the scene. Given the 3D model of the object, the goal is to find a correct transformation (six degrees of freedom – 6DOF) of the 3D model in the point cloud reflecting the current scene. A 3D model can be obtained either by 3D reconstruction or based on the CAD model of the object, which is transformed into a point cloud during configuration. The 3D Object Recognition module is based on the Randomized Global Object Localization Algorithm (RANGO) [24]. Resulting object detections are used to plan collision-free robot movement paths for object manipulation. The accuracy of the 3D object recognition approach described above greatly depends on the sensor data quality and on the sizes of the objects of interest.

**Collision-Free Path Planning:** The results of the Object Recognition and Localization system are used to plan and calculate collision-free robot manipulation paths [2] to enable handling of the detected objects. Based on predefined grasp as well as deposit points on the CAD model of the objects, the manipulation planner determines how the object can be grasped. All object localizations as well as the available workspace environment data are considered for collision checks.

**Robot Interfaces:** To facilitate communication with the robotic system, the XRob framework provides a uniform communication interface, which can be extended in a Plug-In like fashion to support robotic systems of different vendors.

**Application Development:** The XRob software framework provides an intuitive user interface for application development, which includes an interactive programming environment, and software modules to simulate and visualize robotic movement paths as well as data acquisition via sensors.

## 3. Programming by instrumented tools

Human Robot Collaboration (HRC) takes place in safety fence less robot systems which can be useful when full automation is not economically viable. Learning by Demonstration (LbD) is of interest in particular because (re-)programming takes place more frequently in HRC systems – that are typically applied for smaller batch size production when full automation is not viable. The work in [9] provides a concise taxonomy for high and low level LbD and distinguish trajectory level demonstration from symbolic encoding methods. The work in [10] presents two real world production use cases with contrasting requirements in process complexity, required process steps and product geometry or weight. The authors in [11] apply LbD on trajectory level for this application and introduce reusable and task-agnostic motion primitives for assessing the outcome of force-interaction robotic skills. Trajectory demonstration by hand guidance of a process tool mounted on the robot could be less intuitive than demonstrating it with a hand guided process tool (that is instrumented with sensors to measure process data) because it is not embodied with the human demonstrator. As an intermediate approach between trajectory encoding and symbolic encoding we propose to derive the parameterization of macro based skills from a trajectories demonstrated with an instrumented tool rather than from numerical input of process parameters. Human Robot Collaboration (HRC) takes place in safety fence less robot systems which can be useful when full automation is not economically viable. Learning by Demonstration (LbD) is of interest in particular because (re-)programming takes place more frequently in HRC systems – that are typically applied for smaller batch size production when full automation is not viable. The work in [9] provides a concise taxonomy for high and low level LbD and distinguish trajectory level demonstration from symbolic encoding methods. The work in [10] presents two real world production use cases with contrasting requirements in process complexity, required process steps and product geometry or weight. The authors in [11] apply LbD on trajectory level for this application and introduce reusable and task-agnostic motion primitives for assessing the outcome of force-interaction robotic skills. Trajectory demonstration by hand guidance of a process tool mounted on the robot could be less intuitive than demonstrating it with a hand guided process tool (that is instrumented with sensors to measure process data) because it is not embodied with the human demonstrator. As an intermediate approach between trajectory encoding and symbolic encoding we propose to derive the parameterization of macro based skills from a trajectories demonstrated with an instrumented tool rather than from numerical input of process parameters.

### 3.1 Hardware setup

The instrumented tool (see Fig. 3) consists of a power tool, a HTC-Vive™ lighthouse pose tracking system,[1] an ATI-FT9720 Delta SI-330-30 force torque sensor[2] that decouples an L-shaped chassis from an U-shaped handle. The force torque sensor (FTS) is calibrated to measure wrench of ±330 N and ±30 N m. The HTC-Vive™ tracking systems expected accuracy is reportedly better than 2 mm root mean square [12], [13]. The data recorder can be armed with a switch arms. Power tool rotation and data recording can be triggered at the same time. The instrumented tool is used in a workbench that provides a KUKA IIWA 14 R820 robot.[3] The Lighthouse-tracking base-stations are provided on tripod and aluminum profiles.

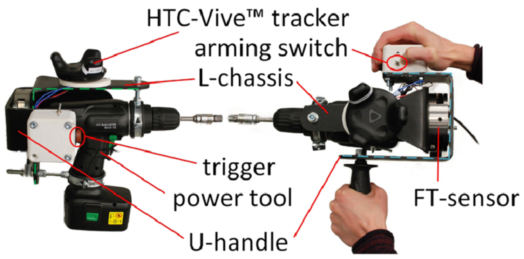


**Fig. 3. Instrumented Tool – hardware setup**

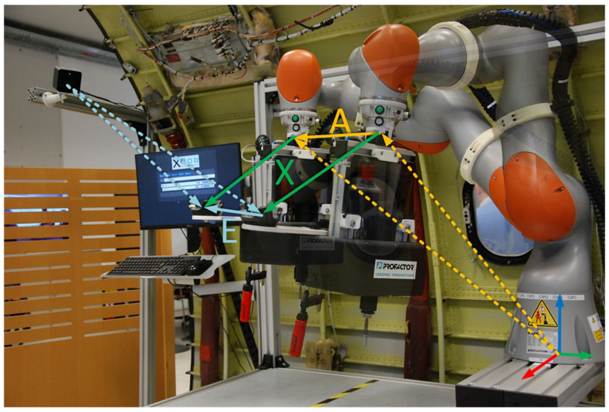


**Fig. 4. Instrumented Tool – hand eye calibration problem**

### 3.2 Registration of the tracking system

The registration of the lighthouse tracking system with the robot can be formulated as hand eye calibration problem (see Fig. 4). The approaches in [14] and [15] provide solutions for solving the equation of type (1) where $\mathbf{A}_i$ indicates the i-th differential flange-frame transformation, $\mathbf{X}$ denotes the transformation of the tracker to the flange frame and $\mathbf{E}_i$ denotes the differential transformation of the tracker in the camera coordinates.

$\mathbf{A}_i$ and $\mathbf{E}_i$ are calculated according to (2) and (3) by selection of the elements (with indices $j$ and $k$) of the measurement pairs $\mathbf{P}_x$ (4). The expected precision of the calibration $\mathbf{T}_{robot\_base} = avg(\mathbf{T}_{robot\_base,j})$ is calculated as square-norm of the standard deviation (over all measurements $j$) of the translational part of the transformations denoted in (5).

$$\mathbf{A}_i \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{E}_i \tag{1}$$

$$\mathbf{A}_i = \mathbf{T}^{-1}_{robot\_flange,j} \cdot \mathbf{T}_{robot\_flange,k} \tag{2}$$

$$\mathbf{E}_i = \mathbf{T}^{-1}_{base\_tracker,j} \cdot \mathbf{T}_{base\_tracker,k} \tag{3}$$

$$\mathbf{P}_x = (\mathbf{T}_{robot\_flange,x}, \mathbf{T}_{base\_tracker,x}) \tag{4}$$

[1] Vive.com. (2019). VIVE™ – VIVE Tracker. [online] Available at: https://www.vive.com/eu/vive-tracker [Accessed 10 Aug. 2019].

[2] Ati-ia.com. (2019). ATI Industrial Automation: F/T Sensor Delta. [online] Available at: https://www.ati-ia.com [Accessed 10 Aug. 2019].

[3] KUKA AG. (2019). LBR iiwa – KUKA AG. [online] Available at: https://www.kuka.com/en-at/products/robotics-systems/industrial-robots/lbr-iiwa [Accessed 10 Aug. 2019].

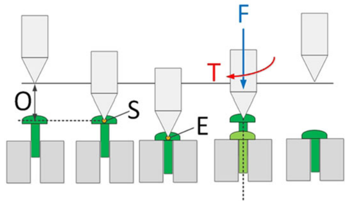


**Fig. 5. Instrumented Tool – process macro**

$$\mathbf{T}_{robot\_base,j} = \mathbf{T}_{robot\_flange,j} \cdot \mathbf{X} \cdot \mathbf{T}_{tracker\_base,j} \tag{5}$$

### 3.3 Process wrench

An unbiased force torque sensor provides calibrated wrench data. The Newton-Euler equations, as shown in (6) and (7) for the instrumented tool can be reordered to calculate sensor forces $\mathbf{f}_S$ and torques $\mathbf{M}_S$ of the externally unloaded (external generalized force $\mathbf{f}_e = 0$ and external generalized torque $\mathbf{M}_e = 0$) instrumented tool which can be used for inertia compensation of the sensor signals that is required to measure process forces only. The mass m (2.7141 g measured on a precision scale) is assumed as point mass. The center of mass (COM) is calculated by considering measured forces and torques as well as the tool center point (TCP) was calculated in similar fashion as (1) was solved. Equations (6) and (7) can be evaluated to calculate the external TCP wrench.

$$\sum \mathbf{f} = m\ddot{\mathbf{r}} + \boldsymbol{\omega} \times m\dot{\mathbf{r}} = m\mathbf{g} + \mathbf{f}_e + \mathbf{f}_S \tag{6}$$

$$\sum \mathbf{M} = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{M}_S + \mathbf{r}_S \times \mathbf{F}_S + \mathbf{r}_e \times \mathbf{f}_e \tag{7}$$

### 3.4 Skill macro & results

The skill macro considers the parameters start-position S, end-position E, offset O as well as process force F and final torque T (see Fig. 5).

The instrumented tool is integrated on top of ROS where all sensor information is provided in relevant topics with 100 Hz update frequency. ROS synchronizes messages so aligned time series of wrench and pose data can be retrieved. After processing the recorded trajectories relevant parameters are extracted from the recorded trajectories (see Fig. 6 – reduced to 1 datapoint every 0.1 s) and provided via web server to the skill/macro based XROB™ run-time system which controls the robot and provides additional functionality like graphical skill/macro based programming with composite skills that provide screwing with sensor based position accuracy compensation.

Figure 7 shows a visualization of the recorded trajectory. Noisy acceleration signals (that occur after the torque clutch limits process wrench from the moment the screw is tight) the measurements are disturbed. A tracking system which is less prone to noisy signals due to vibration is recommended.

Usage of an instrumented tool can help to avoid programming mistakes (e.g. typos) or reduce training requirements for potential users.

## 4. Programming by interactive demonstrations

This paper aims at combining the aspects of LbD, LbP and LbI methodologies in a goal directed assembly process to tackle the problem of easy programming of robotic tasks with the help of a two-phase approach. This work explains the methodology in which the different learning paradigms are planned to be combined to achieve an easy to use programming framework.

**Phase1:** Human operator physically demonstrates the assembly process (AP) to the robotic system by performing a set of activities (interaction with objects). These activities are either demonstrated "hands-free" [8] or using an instrumented tool (Sect. 3. Each activity corresponds to a task in the AP. From the demonstration (LbD), the system captures [7] the semantic relation between the task performed by the human operator and the corresponding consequence

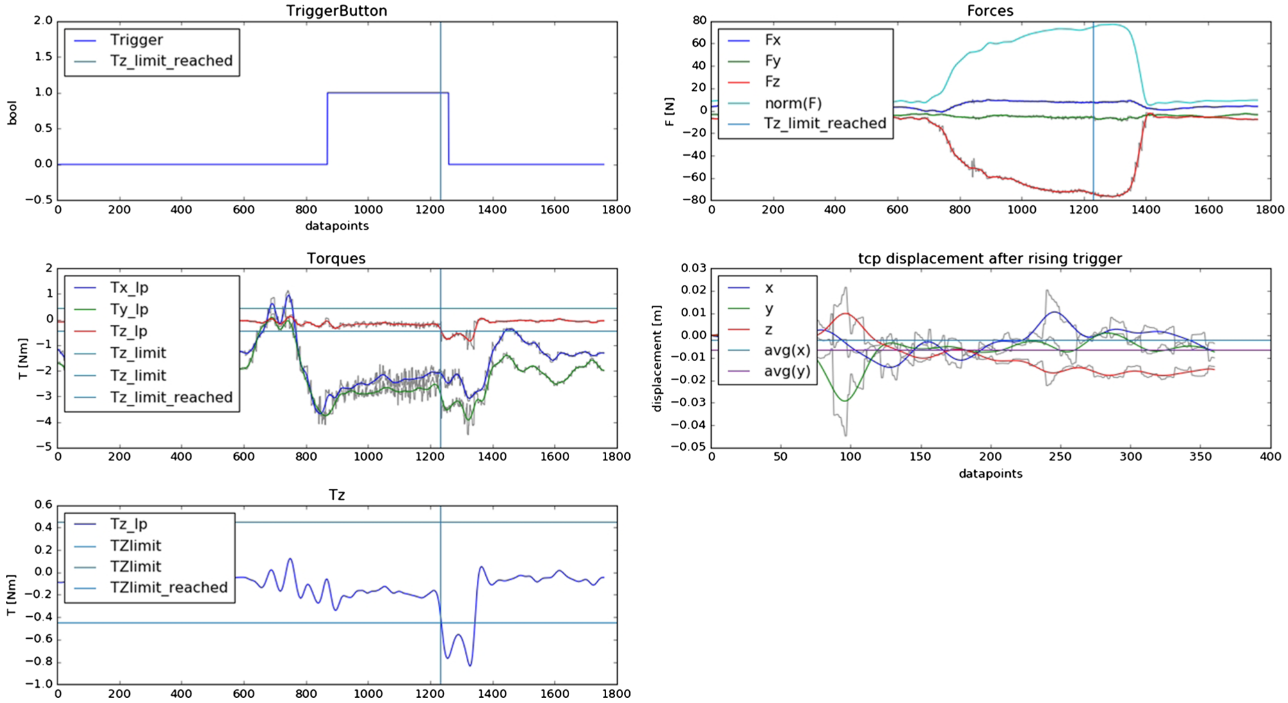


**Fig. 6. Instrumented Tool – measured wrench (Color figure online)**

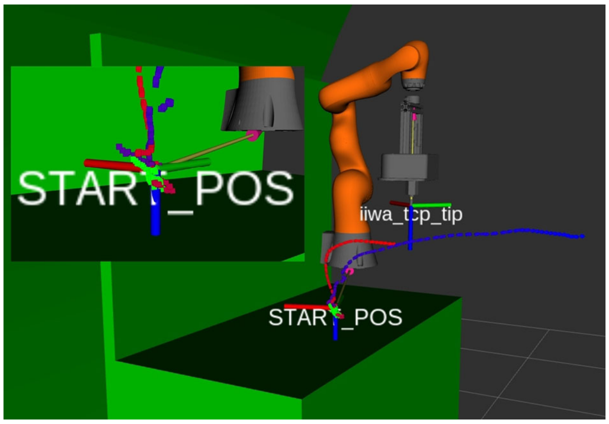


**Fig. 7. Instrumented Tool – trajectory visualization**

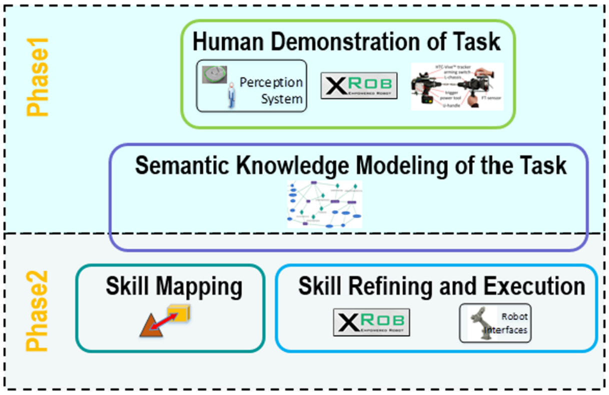

Phase1
Human Demonstration of Task
Perception System
XROB
Semantic Knowledge Modeling of the Task
Phase2
Skill Mapping
Skill Refining and Execution
XROB
Robot Interfaces

**Fig. 8. Architecture of the proposed learning approach**

to the assembly environment [5]. As a result, the knowledge about the sequence of tasks in the assembly process is learned. This creates shared understanding between the human and the robot about the AP at task level. Abstracting the knowledge at task level allows the system to easily learn complex assembly processes with the help of human demonstrations.

**Phase2:** This phase further refines the coarse knowledge obtained in Phase1, to learn process level parameters required for the actual execution of the task by the robot. First, the robot maps (LbP) the tasks from the human domain to that of the robot embodiment. The mapping also takes place at a task level, where the robot tasks are formalized using TFF [1]. The robot extracts the required parameters to execute the tasks in a goal directed fashion [6]. In case of missing parameters, the robotic system queries the human with the help of an interactive GUI (as shown in Sect. 2). When the system has to deal with different similar looking objects, it maps the parameters from previous known objects to the new ones (knowledge transfer [6]) and queries the user for specific modifications. In such cases, the robot queries (LbI) the user only for parameters that vary and does not require the user to re-parametrize the complete task. These new parametrization for similar objects is used in a feedback loop to improve the parametrization [7] of similar cases in the future. The robotic system in Phase1 acquires a fast but general representation of the AP. In Phase2, with the help of intelligent user interactions, the robotic system learns the specific parameters and successfully executes the tasks to complete the AP. This combination could lead to a faster programming phase that is more precise than just demonstrations, and more intuitive than just through a GUI.

The system architecture for the proposed learning approach is shown in Fig. 8.

For the learning architecture to enable easy teaching of an AP to the robot, there are four modules envisioned:

### 4.1 Human demonstration of the task

This module deals with the perceptual aspect. This includes representation of agents' (Robot/Human) activities, where activities range from interaction with objects to simple atomic movements. Recognition of such activities including objects in real-time using state-of-the-art approaches [7], sensor technologies and instrumented tools (as described in Sect. 3)

### 4.2 Semantic knowledge modeling of the task

This involves modeling the knowledge of the assembly process in an action-centric approach for ease of inference [5]. Learning the consequences of the agent's activities and their semantic relation to the assembly process. This step exploits the learning by demonstration methodology to learn the task sequence of the assembly process and abstract the necessary knowledge.

Learning an assembly process by interactive demonstration requires an abstraction of the *knowledge* of an assembly process. In order to develop such a knowledge representation, a modeling language combined with a framework to query and reason about existing data is required. The subsections below provide a brief overview of two state-of-the-art approaches to knowledge modeling and processing, and describe our approach to represent assembly process knowledge. Furthermore, the approach to deriving the selection and parametrization of robot skills in order to imitate the human-demonstrated assembly process through the robot, is explained.

Within the domain of knowledge processing and semantic reasoning, ontology and graph based techniques [17] are common approaches to realize a knowledge representation. The authors in [21] present a thorough overview of existing ontology languages and their applicability for knowledge modeling and information retrieval. A concrete implementation of an ontology based knowledge processing system is given by KnowRob [18] especially for the robotics domain. This framework builds on the Web Ontology Language (OWL, especially OWL description logics) and the Resource Description Framework (RDF) to model the domain knowledge and provides mechanisms for ontology reasoning and inference. The SWI Prolog engine[4] including the Semantic-Web-Library, enables online querying and adaptation of the knowledge base. Hypergraphs[5] are another example commonly used in knowledge processing and AI to express and reason about domain knowledge. This type of graphs are less restrictive in the definition of edges, as they allow to connect multiple vertices over a single edge [19]. Two examples of hypergraph based knowledge processing framework implementations (open-source) are OpenCog [20] and GraknAI [22]. In this work we have chosen the GraknAI framework as an enabler for knowledge modeling and processing.

### 4.3 Skill mapping

The semantic model, targeted in this work, consists of data structures to express an assembly process including a series of *States*, a

[4] Swi-prolog.org. (2019). [online] Available at: https://www.swi-prolog.org/ [Accessed 10 Aug. 2019].

[5] Medium. (2019). Modeling Data with Hypergraphs. [online] Available at: https://blog.grakn.ai/modelling-data-with-hypergraphs-edff1e12edf0 [Accessed 13 Aug. 2019].

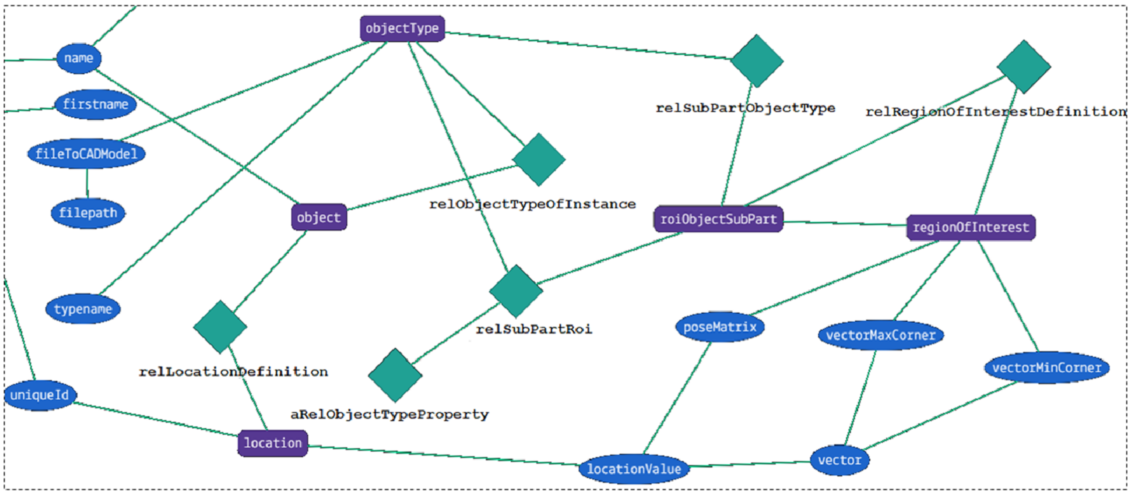


**Fig. 9. Knowledge representation of object types and instance in GraknAI**

set of *Events*, and a set of *Relations*. The States represent the individual assembly steps. An Event is capable, dependent on its definition, to advance the assembly process from a specific state to the next. Relations semantically relate Events and States to form transitions between assembly process states, and the corresponding Events to activate the transitions. A more detailed and formal explanation is given in previous work [16]. Additionally to these fundamental data structures, the semantic model also describes object types and their configurations (e.g. combinations of objects), human and robot skills, and consequences – generated through events – that lead to changes in the environment. Figure 9 depicts a section of the semantic model that describes objectTypes and instances, with given properties and relations.

All concepts (better known as classes), attributes and relations of the semantic model are defined *offline*, thus forming the schema for the representation of knowledge. In the *offline* phase, the knowledge processing framework is populated with relevant process knowledge, including known events and robot skills, including their causal effects. During the *online* phase, e.g. learning phase, queries are issued to the knowledge processing framework in order to generate concrete instances of states, events or objects. These queries are parameterized based on the data, which is delivered by the perception systems that observe the environment during the learning phase. Based on the learned assembly process sequence, the knowledge processing and reasoning system has to answer the question of mapping human skills to robot skills including the parameterization. This problem is solved by a similar representation of effects caused by both human skills (related to detected events) and robot skills, through so-called *consequences*. Consequences describe causal changes in the environment, which include (a) changes in the spatial configuration of objects (e.g. combination of two objects), (b) objects disappearing or appearing newly, and c) displacement of objects. By applying the *concept and relation reasoning functionalities* [23] of the knowledge processing framework, direct relations between human skills and robot skills can be implicitly established, by matching the respective causal effects (consequences). This mapping is realized through rule definitions, which are defined in the semantic model and will be evaluated upon write access to the database. In the given case that the parameterization of an recorded event during learning phase, cannot be mapped to the robot skill, user interaction will be triggered to specify the correct parameterization.

### 4.4 Task refining and execution

The module maps the necessary task to execute to match the present embodiment and status of the robot [1]. Each task is further converted into robot skills using the XRob framework (see Sect. 2) and necessary parameters are extracted. An interactive GUI is also developed that enables an intuitive communication between user and the robot. In case of missing parameters or ambiguities, the robotic system queries the user for feedback to receive the missing parameters with the help of the GUI. However, the system queries the user in an intelligent way posing the questions in a user centric fashion.

## 5. Conclusion

Though learning paradigms for robotic tasks has received extensive attention in the research community, the topic of reducing the programming effort to teach a task to the robot is still open. Different paradigm such Learning by Demonstration (LbD), Learning by Programming (LbP) and Learning by Interaction (LbI) each have their advantages but still fall short in achieving an effective way to easily teach a task to the robot. This work aims to combines these different approaches in a goal-directed fashion to develop a framework that exploits the advantages of these learning paradigms to alleviate the problem of high programming effort. The paper first presents programming framework to easily edit a process workflow (LbP). This is followed with a more detailed description and evaluation about an instrumented tool based learning by demonstration approach. Finally, we present a framework that combines these two learning paradigms in an interactive fashion to reduce the programming effort in teaching a complex task to the robot.

## Acknowledgements

This research is funded by the projects ASKRob, LERN4MRK_II (Austrian Institute of Technology), MMAssist_II (FFG, 858623), Smart Factory Lab and DigiManu (funded by the State of Upper Austria).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

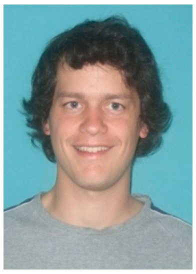
### References

1. Pedersen, M. R., et al. (2016): Robot skills for manufacturing: from concept to industrial deployment. Robot. Comput.-Integr. Manuf., 37, 282–291.

2. Pichler, A., Akkaladevi, S. C., et al. (2017): Towards shared autonomy for robotic tasks in manufacturing. In Proc. FAIM 2017, Modena, Italy.
3. Argall, B. D., Chernova, S., Veloso, M., Browning, B. (2009): A survey of robot learning from demonstration. Robot. Auton. Syst., 57(5), 469–483.
4. Kartoun, U., Stern, H., Edan, Y. (2010): A human-robot collaborative reinforcement learning algorithm. J. Intell. Robot. Syst., 60(2), 217–239.
5. Ramirez-Amaro, K., Beetz, M., Cheng, G. (2017): Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. Artif. Intell., 247, 95–118.
6. Mollard, Y., Munzer, T., Baisero, A., Toussaint, M., Lopes, M. (2015): Robot programming from demonstration, feedback and transfer. In 2015 IEEE/RSJ international conference on intelligent robots and systems, IROS, Hamburg (pp. 1825–1831).
7. Akkaladevi, S. C., Plasch, M., Maddukuri, S., Eitzinger, C., Pichler, A., Rinner, B. (2018): Toward an interactive reinforcement based learning framework for human robot collaborative assembly processes. Front. Robot. AI 5(126). https://doi.org/10.3389/frobt.2018.00126.
8. Akkaladevi, S. C., Plasch, M., Eitzinger, C., Pichler, A., Rinner, B. (2018): Towards a context enhanced framework for multi object tracking in human robot collaboration. In IEEE/RSJ international conference on intelligent robots and systems, IROS, Madrid, Spain, October 1–5, 2018 (pp. 8435–8442).
9. Billard, A., et al. (2008): Robot programming by demonstration. Springer handbook of robotics (pp. 1371–1394). Berlin, Heidelberg: Springer.
10. Ikeda, M., Maddukuri, S., Hofmann, M., Pichler, A., Zhang, X., Polydoros, A., Neugebauer, U. (2018): FlexRoP-flexible, assistive robots for customized production. In Austrian robotics workshop. (p. 53).
11. Zhang, X., Polydoros, A. S., Piater, J. (2018): Learning movement assessment primitives for force interaction skills. Preprint arXiv:1805.04354.
12. Niehorster, D. C., Li, L., Lappe, M. (2017): The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research. i-Perception, 8(3), 2041669517708205.
13. Analysis of valve's 'lighthouse' tracking system reveals accuracy. 14.02.2019 retrieved from https://www.roadtovr.com/analysis-of-valves-lighthouse-tracking-system-reveals-accuracy/.
14. Tsai, R. Y., Lenz, R. K. (1989): A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Trans. Robot. Autom., 5(3), 345–358.
15. Park, F. C., Martin, B. J. (1994): Robot sensor calibration: solving AX = XB on the Euclidean group. IEEE Trans. Robot. Autom., 10(5), 717–721.
16. Akkaladevi, S. C., Plasch, M., Pichler, A., Rinner, B. (2016): Human robot collaboration to reach a common goal in an assembly process. In STAIRS (pp. 3–14).
17. Ehrlinger, L., Wöß, W. (2016) Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS), 48.
18. Tenorth, M., Beetz, M. (2017): Representations for robot knowledge in the KnowRob framework. Artif. Intell., 247, 151–169.
19. Goertzel, B. (2006): Patterns, hypergraphs and embodied general intelligence. In The 2006 IEEE international joint conference on neural network proceedings (pp. 451–458). New York: IEEE.
20. Hart, D., Goertzel, B. (2008): Opencog: a software framework for integrative artificial general intelligence. In AGI (pp. 468–472).
21. Munir, K., Anjum, M. S. (2018): The use of ontologies for effective knowledge modelling and information retrieval. Appl. Comput. Inf., 14(2), 116–126.
22. Klarman, S. (2017): Modelling data with hypergraphs – a closer look at GRAKN.AI's hypergraph data model. Retrieved from https://blog.grakn.ai/modelling-data-with-hypergraphs-edff1e12edf0.
23. GraknAI – A knowledge graph (2019): Retrieved from https://grakn.ai/about.
24. CANDELOR: A computer vision library for 3D scene interpretation. Retrieved from http://candelor.com.

## Authors

**Sharath Chandra Akkaladevi** received his Master degree (with distinction) from the Alpen-Adria University of Klagenfurt in 2012. During which he also worked for 6 months in Stuttgart, Germany at Robert Bosch on driver assistive systems. After joining Profactor GmbH in 2012, he has been working as a researcher in the field of robotics and assistive systems. Since 2015, he is a PhD candidate at the Institute of Networked and Embedded system, Alpen-Adria University Klagenfurt. His main research interests include human robot collaboration, learning and reasoning, cognitive vision and activity recognition.



**Matthias Plasch** graduated at University of Technology, Vienna in April 2012 in the field of "Automation- and Control-Engineering". He has been working as scientific researcher at Profactor GmbH since May 2011. His current main research activities include knowledge processing and reasoning for human robot collaboration, cognitive systems. Additional research interests are automation and control of distributed control systems based on the standards IEC 61499 and OPC-UA, as well as intuitive and simplified programming methods applying workflow management technologies.

**Andreas Pichler** is the CTO of Profactor. He received his graduation degree in Electrical Engineering with a strong focus on automation and control engineering (2000) and his PhD (2004) in engineering sciences from Vienna University of Technology. He joined Profactor in 2004. Until 2011 he was responsible for the development of advanced robot systems which are highly flexible to handle small lotsizes in manufacturing. His research interests include 3D vision, assistive systems in industrial environments, collaborative robot systems and advanced manufacturing systems.

**Markus Ikeda** graduated at Johannes Kepler University, Linz in March 2006 in the field of "Mechatronics/Robotics". He has been working as scientific researcher at Profactor GmbH since July 2006. Current main research activities include human machine interfaces for human robot collaboration and process knowledge acquisition for robotic processes (e.g. screwing). Additional research interests are process knowledge deployment applying both traditional path planning as well as machine learning methodology.

**Michael Hofmann**
graduated at University of Applied Sciences, Hagenberg in July 2010 in the field of "Embedded System Design". He has been working as scientific researcher at Profactor GmbH within the "Robotics and Adaptive Systems" department since September 2010. The current research activities human robot cooperation and zero programming of robotic systems. Additional interests are automation and control of distributed control systems based on the industrial standard IEC 61499.