 DOI: 10.3217/978-3-85125-663-5-14

Workflow-based programming of human-robot interaction for collaborative assembly stations*

Roman Froschauer[1] and René Lindorfer[2]

Abstract— **In certain domains manual assembly of products is still a key success factor considering quality and flexibility. Especially when thinking of flexibility traditional, fully automated assembly using specialized robot stations is mostly not feasible for small lot sizes due to high costs for programming and mechanical adaptations. In the last years collaborative robots (cobots) entered the market to broaden the way for robot-assisted manual assembly. The idea was to use the robot for small repetitive tasks at the manual assembly station and keep the human factor for dealing with flexibility. Unfortunately most of the new cobots came with the same programming system as their ancient relatives. Thinking of human-robot collaboration these traditional approaches do not consider the human factor at the assembly station. Therefore, this paper presents a new approach, called *Human Robot Time and Motion* (HRTM) providing a modeling language providing generic basic elements which can be performed by a human worker or a robot. Correspondingly a workflow-oriented programming model and a prototypical development environment featuring BPMN and MQTT is presented.**

I. Introduction

The number of different variants of products offered by companies is increasing [12]. Therefore, the typical lot size decreases resulting in a demand for more flexible and adaptable production facilities [15]. Mass production companies get the same problem as SMEs (small and medium enterprises), many products with a small lot size down to one [13]. The use of industrial robots in such a flexible production is hardly or not at all possible with today's programming methods [25],[9]. Therefore, human-robot-collaborations (HRC) are getting more and more important [3]. At present working steps of human workers and robots can only be modeled separately. A possible method for human workers is MTM (Methods Time Measurement) [4]. Typically, working steps of robots are not modeled at all, but more programmed using offline programming methods. Nevertheless working steps of robots may also be modeled with RTM (Robot Time and Motion) [24]. Generally, working steps of human workers can be much more complex than those of robots. As a result of this MTM provides much more possible movements than RTM. In case of collaborative robots a human worker and a robot perform actions simultaneously. Therefore, the interaction between both should be modeled [5]. Such collaboration between humans and robots has already been researched through simulations [21]. Additionally, a distinction between a work instruction for a human worker or a robot has to be made. In order to enable comprehensive modeling of working steps in an intuitive way a graphical representation is necessary, which also can be used as work instruction itself to guide human workers through their working steps [11]. A robot can be programmed in different ways but the core elements are always a point and orientation which have to be reached [7], [29]. In order to enable unified modeling of human and robot movements, these two different approaches of modeling a sequence of working steps have to be consolidated. Furthermore the communication between all involved workers (human or robots) has to be considered within the workflow.

The visualization of a workflow is just as important as its model. Different styles are common to visualize a workflow or a process for different fields of application [17]. Due to their visualization style, BPMN (Business Process Model and Notation), AD (Activity Diagram) and EPC (Event driven Process Chain) are appropriate for a detailed check. Additional requirements must be met to visualize HRC workflows.

This paper is organized as follows. In the next chapter we discuss relevant state of the art focusing on workflow modeling for industrial assembly purpose. In chapter III we propose a complete modeling hierarchy featuring a meta-model for collaborative assembly workflows and a specific example. In chapter IV a corresponding development environment is presented using a simple Pick&Place scenario. Finally open issues and future work are discussed.

II. State of the Art

Common languages and methods describing workflows can be divided into two groups: (1) The first group of methods deals with the analysis of the workflow itself and focuses on the detailed mapping of the workflow so that the timing/scheduling can be optimized accordingly. (2) The second group focuses on the general visualization of the workflow, whereas timing is only of secondary importance. Those, which have future potential for a workflow description and are well-established in research or industry, are described in this section.

A. Workflow analysis

To describe arbitrary operations of human workers the *MTM* can be used. Its modeling elements are divided into basic movement elements which are applicable for a manufacturing workflow. As a result, each motion sequence

*This work was supported by the Austrian Research Promotion Agency within the 6th COIN - Cooperation and Innovation programme under grant agreement nr. 856362

[1]Roman Froschauer is with the School of Engineering, University of Applied Sciences Upper Austria, 4600 Wels, Austria roman.froschauer@fh-wels.at

[2]René Lindorfer is with the School of Engineering, University of Applied Sciences Upper Austria, 4600 Wels, Austria rene.lindorfer@fh-wels.at

can be precisely described. By setting different attributes, a basic movement element can be adopted to specific requirements/scenarios. These attributes influence the standard duration time and the difficulty of each basic element. Therefore, all workflows can be accurately analyzed and optimized. *MTM* has been continuously developed further. Thus, several versions have been designed which have been optimized for different application domains [1], [4].

Robots are not able to perform the same motions as human workers. In contrast, they have mechanical constraints, limited workspace and intelligence. As a result, the *RTM* method was developed. *MTM* was used as a basis and got accordingly adapted and extended. Consequently, it is possible to precisely model the motions of a robot and compare the robot's and human's performance time. Similar to *MTM*, attributes influence the standard duration time and the difficulty of each basic movement element. Since the publication of *RTM* it was constantly updated. The latest version was extended with mobility elements [23], [24], [20]. Collaborative modeling elements are still missing.

B. Workflow visualization

The Unified Modeling Language (UML) is mainly used in the field of software development. In UML2.0 different diagram styles are defined, the Activity Diagram (AD) is used to visualize how a system realize a particular behavior. The main elements of an activity in the AD are *Action* elements which are connected with *Control Flow* elements. But the granularity of an activity is not defined, therefore no standard *Action* elements are described. Due to this universal description, AD can also be used to model workflows. However, the main focus is on the visualization for a following implementation of the process [17], [26], [27].

In contrast to AD, the method Business Process Model and Notation (BPMN) is specialized in describing business processes. BPMN is currently available in v2.0.2. It is a widely accepted standard and understandable for all involved parties (drafting, implementation and monitoring) of a business process. The elements of BPMN are very similar to AD. But by using BPMN, the focus is on the modeling of the process for describing the process run through the company and not on a software implementation [17], [10], [19].

With the use of an Event Driven Process Chain (EPC), the focus is placed on occurring events which trigger a function that generates an output event again. Thus, EPCs provide a dynamic view of processes. In EPC only three elements are defined: Functions, Events and Logical connectors. Due to this simple definition of semantic and syntax the models are very flexible, but it also leaves room for interpretation [17], [31], [18], [14].

C. Discussion

In order to model a workflow for a human-robot-collaboration, the method used must satisfy certain requirements. Firstly, it must provide a set of basic elements performable by humans and robots. This ensures that the worker, human being or robot, can also carry out the work instructions. These basic elements must be abstracted to such an extent that they are independent of whether the worker is a robot or a human being. Furthermore, the sequence and duration of the individual working steps must be definable. In order to carry out a single working step, human workers and robots need different specifications. Therefore, it is also necessary to be able to define the attributes of a working step flexibly. If different variants of a product can be represented in one workflow, an on-the-fly variant-change of the current assembly process shall be possible. It is therefore necessary to be able to define decision points in the workflow. Finally, the method has to support collaborative working steps. Therefore, parallel workflows and communication between these workflows must be possible. Now the methods mentioned above are evaluated on the basis of these requirements.

None of the methods fulfill all requirements. *MTM* and *RTM* have been specially developed for the analysis and optimization of human and robot workflows [4], [20]. For this reason, they provide basic elements and the sequence of the processes is determined by a tabular recording. The lead time of a working step is determined by the definition of particular attributes. Further attributes added do not affect the lead time. *AD*, *BPMN* and *EPC* were developed to model software, business or event processes [17]. These modeling languages are characterized by their high flexibility. This means that attributes can be added flexibly, but that they do not provide any basic elements like *MTM* and *RTM*. The different process paths can be defined with the help of decision points. The sequence of the steps is also specified during modeling, but it is not planned to define a lead time.

Subsequently, we merged MTM with RTM [28] and combined it with a universal modeling approach called ADAPT [16]. The universal modeling approach is intended to allow a shift of programming complexity from the end user to a modeling expert, which is only required for the first modeling of a new domain.

III. Approach

Our generic approach focuses on two major parts of modeling discrete processes (such as robot movements): (1) Type of action be accomplished and (2) the required data for each action. Thus we defined a generic model hierarchy enabling custom-specific modeling of actions and assets. This hierarchy illustrated in Fig. 1 features a Meta-Meta-Model defining the basic elements and their relations. The Meta-Model on top enables process- or domain-specific definition of action elements (e.g. using MTM/RTM elements). The final model finally represents a specific workflow using the domain-specific elements.

A. Meta-Meta-Model: ADAPT

As shown in Fig. 1 the Meta-Meta-Model consists of the two major elements called ***Action***, ***Asset*** and three supporting elements, *Decision*, *Relationship* and *Property*.

- *Action* – This element is used to model any kind of task to be accomplished by a machine or a human. Actions can consist of multiple sub-actions. This is shown in

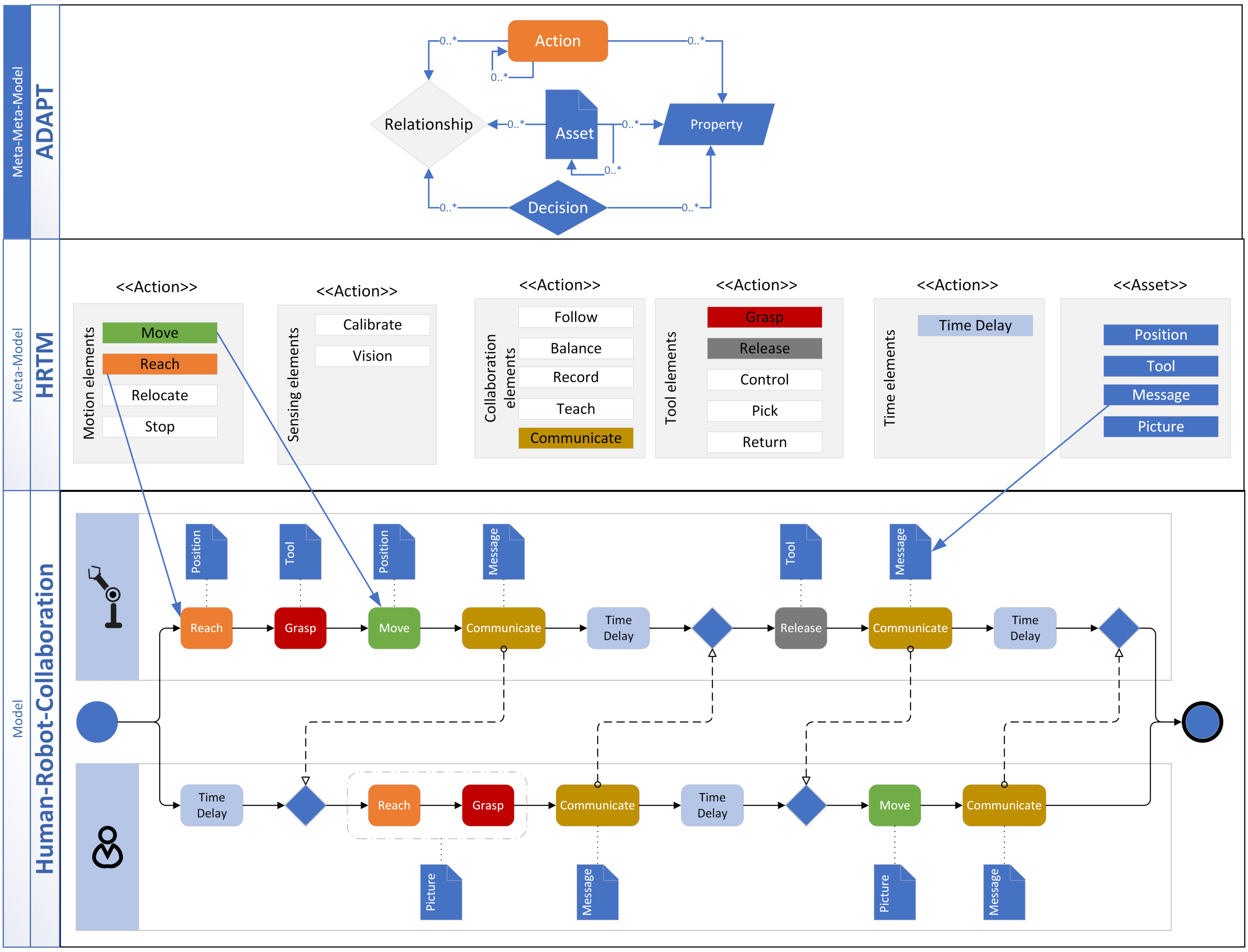

Fig. 1. Model hierarchy (For a better overview relationships, decisions and properties are not shown in full detail)

Fig. 1 with an aggregation of an *Action* to itself. For modeling actions in a generic and reusable manner we use a combination of MTM and RTM methods. An *Action* can be instantiated e.g. as Reach, Grasp, Move, Place, Release, Drill or Screw elements in the Domain Meta-Model defining a set of allowed working tasks. With this Meta-Model it becomes possible to model domain specific models. With corresponding mappings the door is opened to generate platform specific models with implementations for e.g. Reach, Grasp, Move, Place, Release, Drill or Screw (e.g. PLCopen function blocks such as *MC_MoveAbsolute*, *MC_SetPosition*, etc.).

- *Asset* – An *Asset* is a container for any type of information and must include properties. *Assets* can include other *Assets*. This is shown in Fig. 1 with an aggregation of an *Asset* to itself. Typical assets are function blocks containing IEC61131/IEC61499-code (e.g. PLCopen Motion Control), HMI-elements (e.g. buttons, displays, etc.), robot grip positions or work instructions.
- *Decision* – This element models the beginning or fork point of conditional workflows on the basis of offline product configuration data or data gathered from assets at runtime. So it is feasible to model different workflows for various product variants or multiple execution paths. As a result *Decisions* can consume trigger information from any *Asset* (e.g. Camera, Button, Microphone, etc.) for modeling live reactions with various runtime decisions. Depending on this trigger information the *validIf* condition of Relationships described later can be evaluated.
- *Relationship* – Relationship elements can be used to model e.g. aggregation, specialization, predecessor or successor dependencies, can be expanded as desired and consist of a source and a target definition. A relationship element always includes a so called Validity Condition (*validIf*) as presented in [8]. This condition is used to express a kind of variability on the basis of boolean expressions evaluated at design-time or runtime. These boolean expressions are created using boolean variables provided by assets, which are representing logical input

from product configurators or physical input from the modeled process (e.g. Cameras, Sensors or Human-Inputs). If the expression is evaluated true the connected actions are executed or assets are included. This means a conditional action flow on the basis of asset-provided data is possible.

- *Property* – The elements *Decision*, *Asset* and *Action* include *Properties* that describe them in more detail.

Using this core Meta-Meta-Model we now can implement a Meta-Model to be used for modeling collaborative assembly operations.

B. Meta-Model: Human-Robot-Time and Motion

With the previous mentioned ADAPT Meta-Meta-Model the decision-based meta-modeling of collaborative assembly and manufacturing tasks is possible. Therefore, the elements of MTM and RTM are merged and extended by collaborative elements and additional tool elements. In Fig. 1 the 17 defined or instantiated Actions of the so called Human-Robot-Time&Motion approach are shown on the Meta-Model level. In contrast to MTM and RTM method, the HRTM elements do not stipulate a coding or timing assignment. Instead, freely definable Properties can be added to the HRTM elements (i.e. Actions).

C. Modeling elements

Combining MTM/RTM and extending them with collaborative elements results in 17 elements that can be organized in five groups. (1) *Motion elements* represent the movements of a robot or a worker. These include a motion of the arm and a position change, but also the stop of a motion. To describe an action which needs a tool, (2) *Tool elements* are used. In the HRTM approach a tool is defined as an object, which enhance the abilities of the worker or allows the worker to perform a special action. It could be an active tool, like a power screwdriver, or a passive tool, like a stamp. If an interaction with the environment is required, which is not a result of another element, the (3) *Sensing elements* are used. With the help of HRTM collaborative workflows can be modeled. Therefore, (4) *Collaboration elements* are needed for an interaction between two workers. To be able to model a continuous work flow, (5) *Time elements* are used to describe waiting times. The transitions between these action elements are modeled with *Decisions*. In addition, *Actions* as well as *Assets* can be constrained with optional or mandatory relationships, e.g. a Grasp action may also require a Release action.

D. Model: Pick&Place Workflow

For easier modeling of workflows with our HRTM model we use a BPMN-based view model and a corresponding editor. The elements of HRTM are represented with the BPMN element *Activity* and are linked with *Sequence Flow* elements. As defined in BPMN, each workflow starts with a *Start event* and the last element is an *End event*. Additional information can be linked with an *Association* to the inserted HRTM element. Therefore, it is possible to define all necessary data for an action regardless of whether a human worker or a robot executes it.

Action	Group	Description
Reach	Motion	Motion of the hand-arm-system without load
Move	Motion	Move handling of a load
Stop	Motion	Stop of a motion
Relocate	Tool	Repositioning of the worker/robot in space
Control	Tool	Controlling a tool
Grasp	Tool	Indicates that the gripper of a robot or the hand of human worker holds an object
Release	Tool	The gripper of a robot or the hand of human worker is free
Pick	Tool	Pick up a tool from a tool magazine
Return	Tool	Return of a tool into a tool magazine
Calibrate	Sensing	Robot must be calibrated at its current position or a human worker has to orientate himself at the working place
Vision	Sensing	Automated recognizing, identifying, localizing objects
Follow	Collaboration	Mirroring of other movements in general
Balance	Collaboration	Hold the position within the current circumstances
Record	Collaboration	Recording a path, e.g. during teaching (special case of Follow)
Teach	Collaboration	Information exchange from the senior human worker to the novice/robot
Communicate	Collaboration	Information exchange between two or more workers/robots
Time Delay	Time	Wait while the other partner performs an action

TABLE I
HRTM MODELING ELEMENTS REPRESENTED AS ADAPT ACTIONS

Considering the example in chapter IV, the view model of a simple “Pick&Place” workflow is shown in a BPMN-based editor (see Fig. 3). The goal of the example workflow is to place a part on a defined position. Decomposing of a typical human workflow results in a *Reach* element to determine the current position of the part. After reaching the part, it has to be grasped using a *Grasp* element and subsequently placed to the target position with the help of the *Move* element and finally released using a *Release* element. In order to model start and end of a task corresponding *Start-* and *End-events* have to be inserted. Due to the use of sequence flows, the sequencing of the four elements is defined arbitrary.

After modeling the basic sequence of actions the required data to perform these actions has to be assigned by adding source positions, work instruction, etc. Typically human workers are not thinking in HRTM elements, but are automatically combining them to complex tasks. Therefore, our approach supports aggregation of elements in order to create complex tasks and enables a reuse of them. Certain actions such as the *Reach* and *Move* element needs a position and the *Grasp* and *Release* element the corresponding tool. This required information is modeled as Assets and appended to the elements with *Association* lines.

IV. Implementation

This section presents an universal workflow modeling environment WORM implementing the ADAPT-Meta-Meta model. It enables the definition of custom meta-models and supports the specification of custom model elements and their relationships, i.e. custom domain specific languages for graphical modeling. The environment features (1) a tree-based meta-modeling editor, (2) a BPMN-based model/workflow editor and (3) a corresponding runtime-engine, which is able to execute the created workflow model.

A. Meta-Modeling Editor

The meta-modeling editor, called Architect, enables the creation of an ADAPT-based, customer-specific modeling language. In our presented example we created a HRTM model featuring 17 actions and corresponding relationships and assets. Each action now has several relationships such as "Reach includes Picture", "Grasp includes Picture", etc. On the basis of the rules (i.e. grammar) defined within the Architect, several models can be implemented and verified.

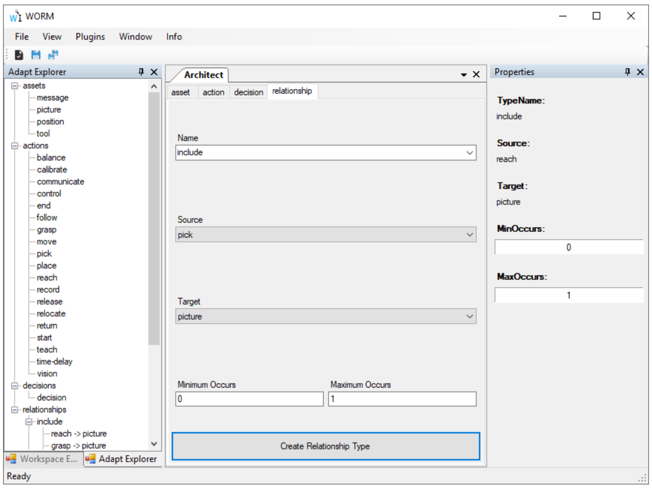

Fig. 2. Creation of the HRTM based Meta-Model.

B. Modeling Editor

In order to create collaborative workflows the user has to model the workflow itself in a sequential way, whereas the user can choose from the HRTM actions modeled in the Meta-Model. In a next step each action may have optional or mandatory relationships to other actions or assets. For example a grasp action requires either a position or in case of an installed vision system a picture of the part to be grasped. The vision system itself may also be linked to the asset picture etc. Continuing this process a complete workflow including all necessary data, e.g. positions, tools, step files, pictures and even worker profiles, is modeled. Thinking of worker profiles the approach enables to assign differently skilled worker or robots to each action. In case of a robot is assigned to an action all assets may be sent directly to the robot by the runtime system (explained below).

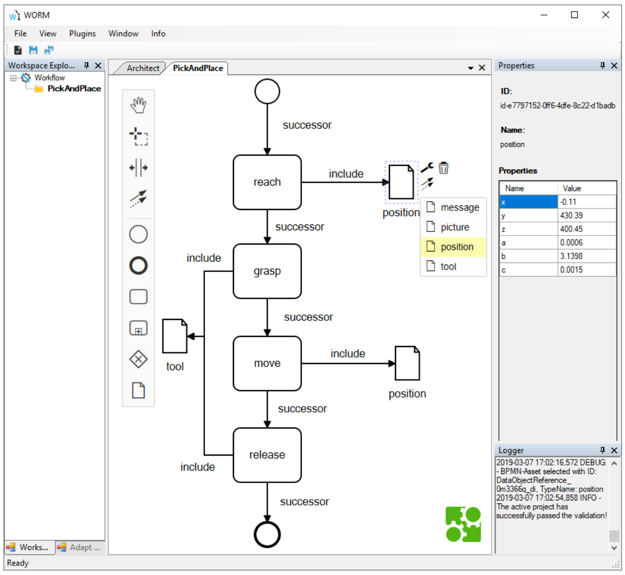

Fig. 3. Creation of a HRTM-based workflow model.

C. Runtime Engine

In order to execute workflows created with our modeling tool a first prototypic runtime engine has been implemented. The workflow, modeled by the Workflow Modeler, can be exported into an XML-representation which is then loaded by the runtime engine. The engine steps through the actions and loads the associated assets from a local asset server (via URL). The assets are then pushed into corresponding MQTT channels [2], to which the devices (i.e. in our showcase the UR10) are subscribed. By this means the robot is supplied with target positions and commands by the runtime engine. As a result one or more robots or any kind of device (such as PLCs with a corresponding listening application) can be programmed using high level workflow modeling. The engine itself is highly customizable in that way which kind of reaction shall be generated during executing the workflow. For our showcase we implemented a MQTT publisher, whereas an OPC UA [22] connector is currently under work.

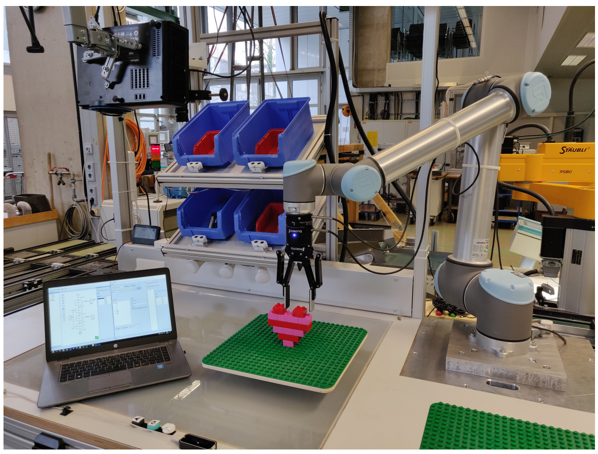

Fig. 4. Demonstration setup.

V. Conclusion and future Work

With the presented modeling approach and its corresponding engineering environment it is possible to model collaborative workflows for humans and robots. As shown in chapter IV, by associating appropriate data to the individual tasks it is possible to create a complete set of data required for programming a robot. For more complex workflows all elements of the HRTM approach as defined in Table I are required. One of the next steps is to define a corresponding minimum set of data (e.g. Assets) for each HRTM Action. This is to ensure that every Action is executable and it is able to perform product specific tasks, depending on constraints such as product dimensions, etc.

When modeling a collaborative workflow, the decision points between the actions must be modeled explicitly in order to support a seamless interaction between humans and robot workers. A human worker automatically switches to the next step as soon as the current step is completed. However, a robot needs specific trigger information to perform the corresponding command at the right time. These triggers must be defined so that the execution of a workflow works correctly on different systems.

For a created HRTM workflow to become reusable, it must be stored in an appropriate way. Currently the model is stored in our ADAPT model hierarchy. In order to enable direct interaction with upcoming toolsets an XML-based transformation into AutomationML may be investigated in future [6]. Furthermore, on the basis of approaches for autonomous assembly program derivation as presented by Thomas [30], this approach may be extended with the possibility of deriving HRTM models out of CAD data and bills of materials.

Finally, with the help of the presented approach an universally readable workflow can be modeled and executed on various devices.

References

[1] W. Antis, J. Honeycutt, and E. Koch, *The basic motions of MTM*. Maynard Foundation, 1973.

[2] A. Banks and R. Gupta, "MQTT version 3.1.1," *OASIS Standard*, 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[3] K. Beumelburg, "Fähigkeitsorientierte montageablaufplanung in der direkten mensch-roboter-kooperation," Ph.D. dissertation, Universität Stuttgart, 2005.

[4] R. Bokranz and K. Landau, *Produktivitätsmanagement von Arbeitssystemen: MTM-Handbuch*. Schäffer Poeschel Stuttgart., 2006.

[5] L. G. Christiernin, "How to describe interaction with a collaborative robot," in *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*. Association for Computing Machinery (ACM), 2017.

[6] Common Working Group of AutomationML e.V and eCl@ss e.V., "Whitepaper: AutomationML and eCl@ss Integration," 2017, V 1.0.1.

[7] B. Denkena, H. Wörn, R. Apitz, R. Bischoff, B. Hein, P. Kowalski, D. Mages, and H. Schuler, "Roboterprogrammierung in der fertigung," *Springer*, vol. 9, pp. 656–660, 2005.

[8] D. Dhungana, P. Grünbacher, and R. Rabiser, "The dopler meta-tool for decision-oriented variability modeling: a multiple case study," *Automated Software Engineering*, vol. 18, no. 1, pp. 77–114, Mar 2011.

[9] P. R. Engelhardt, "System für die rfid-gestützte situationsbasierte produktionssteuerung in der auftragsbezogenen fertigung und montage," Ph.D. dissertation, Technische Universität MÜnchen, 2015.

[10] O. M. Group, *Business Process Model and Notation (BPMN), Version 2.0.2*, Object Management Group Std., Rev. 2.0.2, January 2014.

[11] A. Gupta, D. Fox, B. Curless, and M. Cohen, "DuploTrack: A Real-time System for Authoring and Guiding Duplo Block Assembly," in *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. ACM Press, 2012.

[12] S. Hu, X. Zhu, H. Wang, and Y. Koren, "Product variety and manufacturing complexity in assembly systems and supply chains," *CIRP Annals*, vol. 57, no. 1, pp. 45 – 48, 2008.

[13] N. Keddis, G. Kainz, A. Zoitl, and A. Knoll, "Modeling production workflows in a mass customization era," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, March 2015, pp. 1901–1906.

[14] G. Keller, M. Nüttgens, and A. Scheer, *Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)"*, ser. Institut für Wirtschaftsinformatik Saarbrücken: Veröffentlichungen des Instituts für Wirtschaftsinformatik. Institut für Wirtschaftsinformatik, 1992.

[15] J. Krüger, T. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals - Manufacturing Technology*, vol. 58, no. 2, pp. 628–646, 2009.

[16] R. Lindorfer, R. Froschauer, and G. Schwarz, "Adapt - a decision-model-based approach for modeling collaborative assembly and manufacturing tasks," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, July 2018, pp. 559–564.

[17] B. List and B. Korherr, "An evaluation of conceptual business process modelling languages," in *Proceedings of the 2006 ACM Symposium on Applied Computing*, ser. SAC '06. New York, NY, USA: ACM, 2006, pp. 1532–1539.

[18] P. Loos and T. Allweyer, "Object-orientation in business process modeling through applying event driven process chains (epc) in uml," in *Proceedings Second International Enterprise Distributed Object Computing (Cat. No.98EX244)*, Nov 1998, pp. 102–112.

[19] J. Mendling and M. Weidlich, Eds., *Business Process Model and Notation - 4th International Workshop*. Springer, 2012.

[20] S. Y. Nof, *Handbook of Industrial Robotics*, 2nd ed. John Wiley & Sons, Inc., 1999.

[21] J. Novikova, L. Watts, and T. Inamura, "Modeling human-robot collaboration in a simulated environment," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts - HRI'15 Extended Abstracts*. Association for Computing Machinery (ACM), 2015.

[22] "OPC UA Specification," OPC Foundation, 2015. [Online]. Available: https://opcfoundation.org/developer-tools/specifications-unified-architecture

[23] R. L. Paul and S. Y. Nof, *Human and Robot Task Performance*. Boston, MA: Springer US, 1979, ch. 2, pp. 23–50.

[24] R. P. Paul and S. Y. Nof, "Work methods measurement—a comparison between robot and human task performance," *International Journal of Production Research*, vol. 17, no. 3, pp. 277–303, 1979.

[25] M. Rickert and A. Perzylo, "Industrieroboter für KMU - Flexible und intuitive Prozessbeschreibung," *Industrie 4.0 Management*, pp. 46–49, 2016.

[26] C. Rupp, J. Hahn, S. Queins, M. Jeckle, and B. Zengler, *UML 2 glasklar*. Hanser Fachbuchverlag, 2005.

[27] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and P. Wohed, "On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling," in *Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modelling - Volume 53*, ser. APCCM '06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 95–104.

[28] D. Schönberger, R. Lindorfer, and R. Froschauer, "Modeling workflows for industrial robots considering human-robot-collaboration," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, July 2018, pp. 400–405.

[29] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter*, ser. Springer-Lehrbuch. Springer Berlin Heidelberg, 2013.

[30] U. Thomas, *Automatisierte Programmierung von Robotern für Montageaufgaben*, Prof. Dr.-Ing. Friedrich M. Wahl, Ed. Shaker Verlag, 2008.

[31] W. van der Aalst, "Formalization and verification of event-driven process chains," *Information and Software Technology*, vol. 41, no. 10, pp. 639 – 650, 1999.