Enabling End-Users to Deploy Flexible Human-Robot Teams to Factories of the Future

Dominik Riedelbauch[1], Johannes Hartwig[1] and Dominik Henrich[1]

***Abstract*— Human-Robot Teams offer the flexibility needed for partial automation in small and medium-sized enterprises (SMEs). They will thus be an integral part of Factories of the Future. Our research targets a particularly flexible teaming mode, where agents share tasks dynamically. Such approaches require cognitive robots with reasoning and sensing capabilities. This results in hardware maintenance demands in terms of sensor calibration. In contrast to intuitive end-user programming, system setup and maintenance are rarely addressed in literature on robot application in SMEs. In this paper, we describe a prototype software toolchain that covers the initial setup, task modelling, and online operation of human-robot teams. We further show, that end-users can setup the system quickly and operate the whole toolchain effortlessly. All in all, this work aims to reduce the concern, that deploying human-robot teams comes with high costs for external expertise.**

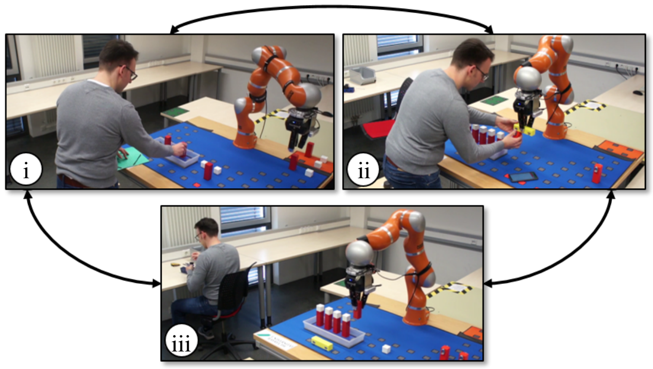

Fig. 1. Flexible Human-Robot Teams can switch between cooperation (i), collaboration (ii) and coexistence (iii) dynamically.

I. BACKGROUND AND RELATED WORK

Robots have started to emerge from potentially hazardous tools that need to be locked behind fences to teammates working hand in hand with humans. Fence-less lightweight robots suit the needs of SMEs, where flexibility to produce varying products with small lot sizes is a key requirement [1]. A major part of these enterprises is already using lightweight robots, or planning to do so within few years [2]. However, humans and robots merely coexist in recent applications rather than forming symbiotic teams that share work and utilize individual strengths of humans and robots [3]. The FLEXCOBOT[2] project seeks to close this gap.

In contrast to numerous works on static human-robot task sharing (e.g. [4][5]), FLEXCOBOT is based on *dynamic task allocation* through online, iterative decisions. To this end, the approach grants decision making authority to all teammates and results in flexible human-robot teams. In particular, our notion of a *flexible team* is characterized by dynamic transitions between three teaming modes (Fig. 1): (i) **Cooperation**: In this mode, human and robot work efficiently by carrying out sub-tasks of the same task in parallel. (ii) **Collaboration**: In contrast to the loose coupling of cooperation, partners work on the same sub-task during collaboration. Physical contact, potentially transmitted via parts to be handled jointly, is intended. (iii) **Coexistence**: Transitioning into coexistence allows workers to flexibly handle urgent intermediate tasks (e.g. handling a delivery of goods or supplying the workstation with material) by leaving the workbench temporarily. Our approach provides the robot with sufficient autonomy to keep working in the meantime. Humans may decide to re-join into cooperation at any time. The robot is equally able to initiate mode transitions, e.g. by actively calling an absent partner for help within a collaborative sub-task. More details on this task allocation method based on skill interaction categories and human-aware world modelling are given in our previous publications [6] and [7]. Video clips are available online[1].

Although flexibility is a key requirement, this feature is not the only concern that must be addressed when designing robot systems for Factories of the Future – the cost factor must also be taken into account, especially with regard to the perceived lack of in-house programming expertise and expected personnel expenditure for external experts [1][2]. This problem has been approached by novel methods for intuitive end-user programming of industrial robots, which are often based on task modelling via graphical user interfaces (e.g. [8][9][10]). These methods reduce the need for external programmers, as they enable the existing workforce to teach the system. Flexible human-robot teaming poses additional requirements on task modelling. A suitable task model must encode potential parallelism of sub-tasks for efficient cooperation. It must moreover supply robots with means to observe and understand task progress. We have previously shown, that the advantages of graphical end-user programming can be transferred to commissioning of flexibly shared tasks in spite of these increased requirements [11].

The issue of understanding task progress brings us to an additional problem that must generally be dealt with

[1] The authors are with the Chair for Applied Computer Science III (Robotics and Embedded Systems), University of Bayreuth, D-95440 Bayreuth, Germany; {dominik.riedelbauch | johannes.hartwig | dominik.henrich}@uni-bayreuth.de;

[2] This project has partly been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696/20 FlexCobot.

[1] http://robotics.uni-bayreuth.de/projects/flexcobot/

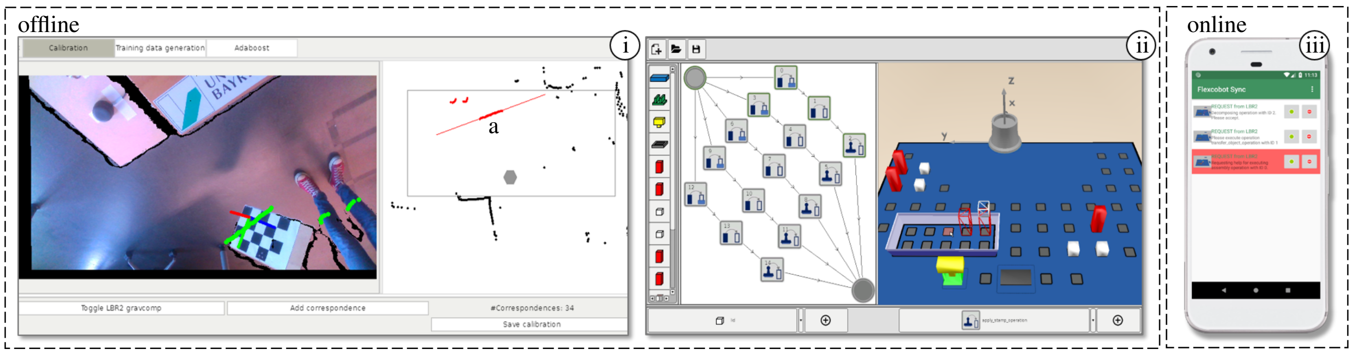

Fig. 2. Our software toolchain consists of tools for calibration (i), task modelling (ii) and human-robot communication (iii) that support the offline and online phase of flexible human-robot teaming.

when adapting robots to new tasks: Smart robot assistants do not follow a program thoughtlessly, but also use sensors to perceive and react to their environment. Usually, these sensors need to be calibrated to the robot to provide a unified coordinate frame, e.g. for reasoning. Having in mind the cost argument that motivates intuitive programming, this step should also be feasible for end-users. The importance of user-friendly ways to calibrate a RGB-D vision system has previously been stated by Paxton et al. in the context of intuitive programming [9], and by Rückert et al. [12] for the field of human-robot collaboration. A convenient one-click solution for calibrating an industrial manipulator to a multi-camera system for safe human-robot coexistence has been proposed by Werner et al. [13]. Overall systems including the calibration step are however not addressed in recent literature on teaming with dynamic task sharing (e.g. [14][15]).

Combining the aforementioned ideas and requirements, the contribution of this short paper is as follows: We will first describe our prototype software toolchain for flexible human-robot teaming in Section II. This toolchain integrates support of end-users throughout all stages of system operation, from the offline steps of initial calibration and intuitive task modelling to online worker support and human-robot communication. The contribution lies not primarily in the composition of these software tools – we rather use the prototype as a basis for showing, that deploying flexible human-robot teams is feasible for non-expert users within a short timespan (Section III). To the best of our knowledge, no comparable overall system for teaming has yet been considered from this point of view in existing literature.

II. A SOFTWARE TOOLCHAIN FOR FLEXIBLE TEAMING

The structure of our software toolchain is shown in Fig. 2. The *offline phase* covers the steps of calibration (i) and task modelling (iii). Based on the resulting task models, human and robot may work as a team in the *online phase*. This phase is supported by a smartphone app for human-robot communication (iii). A detailed view on the individual steps will be given hereinafter. The focus will lie on the technical details of calibration, as this step complements our previous work on task modelling [11] and shared task execution [6][7].

Calibration: The FLEXCOBOT system makes decisions based on sensor data from two sources: An eye-in-hand camera mounted near the robot hand is used to maintain a symbolic world model by recognizing parts in the workspace. This avoids the occlusions that cameras in fixed positions might suffer from. Of course, humans may manipulate previously detected parts, while they are out of sight for the robot – inspired by the process of human forgetting, the system deals with partial observability by losing trust in parts, when they enter the sphere of influence of any human [6]. To this end, humans are tracked in the data provided by a 2D LIDAR sensor near the workbench. Determining, whether humans can access certain objects in the world model requires part and human positions to be known in the same coordinate frame. This brings us to the following two-staged multi-sensor calibration problem:

In order to place parts perceived by the camera in the coordinate frame W of the robot world model, we first need to know the homogenous extrinsic calibration matrix $X \in \mathbb{R}^{4\times4}$ between tool centre point (TCP) and camera frame (cf. Fig. 3). This problem of eye-in-hand calibration is well-known and commonly solved by observing a calibration pattern from N different camera poses. Then, X can be optimized to satisfy

$$A_i X B_i = A_j X B_j \Leftrightarrow A_{ij} X = X B_{ij}, \qquad (1)$$

where $A_{ij} = A_j^{-1} A_i$, $B_{ij} = B_j B_i^{-1}$, $i, j \in \{1, ..., N\}$, $i \neq j$. B_k is known from robot forward kinematics. The pattern position A_k in the camera frame can be directly calculated from camera images since we use a RGB-D camera. An overview of approaches to solve this optimization problem is given by Shah et al. [16]. We used the dual quaternion approach proposed by Daniilidis [17] and refined the result with an iterative non-linear least squares optimization. Corresponding A_k and B_k can be collected using a graphical user interface that moves the robot to predefined poses upon one single mouse click, and starts the optimization afterwards. The calibration pattern is shaped to fit the robot base segment to ensure appropriate placement.

We furthermore need to transform human positions detected by the laser range finder into the world model coordinate frame W. As soon as X is known, the extrinsic LIDAR calibration matrix $Y \in \mathbb{R}^{4\times4}$ between the world model and LIDAR data frame is given as

$$Y = C_k X B_k, \qquad (2)$$

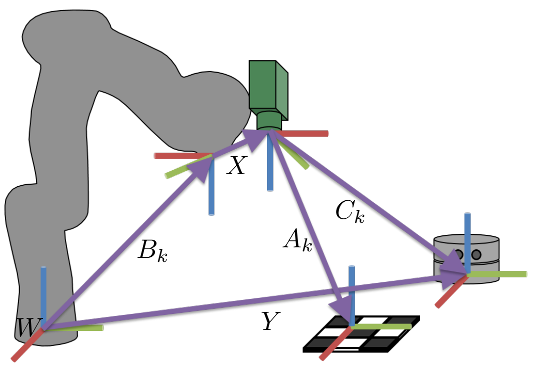

Fig. 3. Solving the calibration problem involves transformations between world frame and robot TCP (B_k), TCP and camera coordinates (X), camera coordinates and LIDAR data frame (C_k), world frame and LIDAR data frame (Y), and the calibration pattern pose wrt. the camera frame (A_k).

for some homogeneous transform $C_k \in \mathbb{R}^{4\times4}$ between camera and LIDAR frame and corresponding robot pose B_k. If we now choose a fixed robot pose B_f, we can reduce the problem to calculating a fixed C_f, i.e by deriving the extrinsic calibration parameters between LIDAR sensor and eye-in-hand camera. This problem can be solved with the method of Zhang and Pless [18]. To this end, the calibration pattern needs to be placed at M different positions in view of the camera, while also being sensed by the LIDAR sensor. At each position $l = 1...M$, the calibration plate is seen as a set of samples P_l forming a line segment in the LIDAR scan (Fig. 2(i)a, red line fit). We know the normal vector N_l of the plate from the RGB-D camera image. All samples $p \in P_l$ must lie on the plane defined by N_l after projecting them into the camera coordinate frame. Thus, $N_l \circ C_f p = \|N_l\|^2$ must hold for all $p \in P_l$. This leaves us with $|P_l|$ equations at each of the M pattern positions. C_f results from optimization over the union of these equations. Using the resulting C_f in Formula 2 provides the searched value of Y. Gathering point-plane correspondences at different pattern positions is supported by a calibration app (Fig. 2(i)). The app allows users to reposition the robot so the camera and the LIDAR sensor can detect the pattern simultaneously. While moving the pattern, the app gathers correspondences, solves for Y repeatedly, and projects LIDAR samples into the camera image (green in Fig. 2(i)). Users may stop the calibration process, as soon as the reprojected samples are seen to cover their legs and the plate sufficiently.

Task Modelling: The graphical editor for shared tasks that we presented initially in [11] is the second software component of the toolchain. This editor is based on a robot skill framework inspired by the work of Andersen et al. [19]. So called motor and perception primitives (e.g. `Grasp`, `Place`, `Hold`, `Transfer`, `TriggerCamera`) are grouped into more complex skills by specifying a graph for control and data flow. We addressed the teaming aspect by demanding a well-defined graph structure, so that object-centric, observable preconditions and postconditions can be deduced from parametrized skills automatically. We moreover added *communication primitives* (e.g. `WaitForAcknowledge`) to synchronize the work of human and robot by communicating within a skill. Skills created by experts or system integrators (e.g. `PickAndPlacePart`, `MateParts` in the current prototype implementation) open modelling of more complex tasks to end-users [8]. To this end, our editor adopts ideas from CAD-based robot programming. Parts can be instantiated by placing them within a virtual environment (Fig. 2(ii), right). They are then used as parameters to skills. It is important to notice at this point, that skill parametrization is solely intended to reflect the process to be modelled. We do not assume, that end-users will take into account robot characteristics (e.g. limited reach due to kinematic properties) when specifying the position of objects on the workbench – the resulting issues are resolved by communication during the online execution phase. Each parametrized instance of a skill is represented by a pictogram (Fig. 2(ii), left) referring to the skill type. Users may establish precedence relations among skills by connecting these pictograms. The overall task modelling process results in precedence graphs as known from the assembly planning domain. Future work may thus open the toolchain to task models originating from automated assembly planning [20].

Online Task-Sharing: Precedence graphs created with the task editor are shared with the robot via an XML representation. Pre- and postconditions of skills help the system to estimate task progress by matching conditions against the state of detected objects in the world model. With this knowledge on progress and the aforementioned trust in world model content, our robot teammate can iteratively try to execute skills that are likely to succeed [6]. As stated before, skills in the task model must not necessarily be feasible for the robot, as this would require expert knowledge in the modelling step. Hence, the dynamic task allocation algorithm is moreover able to reason about feasibility by classifying skills into interaction categories [7]. These categories mirror the degree of interaction that is needed to work off some skill: The robot may e.g. execute a skill all by itself, fully delegate it to the human, or enter collaboration. The latter two cases result in a need for communication. Our software toolchain provides a smartphone application for this purpose (Fig. 2(iii)). This is motivated by the facts that smartphones are already ubiquitous and will certainly be at hand for workers in Smart Factories of the Future. Via this app, the robot may e.g. inform its partner about skills that it is not capable of, or ask the human to switch into the collaboration mode via short messages and visual cues.

III. EVALUATION AND RESULTS

In this section, we will first present novel results from the evaluation of the calibration app. A recapitulation of previous experiments on task modelling and execution enables us to discuss the issue of end-users operating the whole toolchain.

Calibration: Hereinafter, we investigate the hypothesis, that our software tools for calibration enable end-users to perform a complex multi-sensor calibration comfortably within a reasonable amount of time. Against the background of industrial applications in SMEs, participants with a technical

	P1	P2	P3	P4
Camera				
Reading Time [min]	0:44	0:40	0:18	0:24
Execution Time [min]	6:42	6:50	6:20	6:27
# of questions asked	0	0	0	0
LIDAR				
Reading Time [min]	1:01	3:16	1:08	2:02
Execution Time [min]	6:50	6:30	5:21	10:13
# of attempts	2	1	1	2
# of questions asked	2	3	1	3
Overall Calibration Time [min]	15:17	17:16	13:07	19:06

TABLE I
USER EVALUATION RESULTS

background were chosen. None of them indicated having prior knowledge on similar calibration procedures. All subjects were supplied with short one-page user manuals on how to use the calibration interfaces outlined in Section II. After reading these manuals, four users executed both calibration steps. We measured the reading time needed to comprehend the manuals and the time needed for each step (Table I). All in all, none of the participants exceeded an overall time of ca. 20min between reading the manual and completing calibration. The one-click camera calibration is straightforward and thus did not raise any questions beyond the manual. It was performed correctly by all users at the first attempt. LIDAR calibration is more complex and demands users to understand how to collect beneficial correspondences and when sufficient precision has been reached. These issues resulted in a low number of questions across all users. Still, all of them succeeded at least at the second attempt.

Task Modelling: Our user evaluation presented in [11] shows, that non-experts can handle skill-based modelling of precedence graphs for pick&place tasks intuitively. We could furthermore show, that complex graphs with up to 84 elements can be modelled in less than 10 minutes after gathering some experience with the editor.

Online Task-Sharing: Preliminary prior results from simulated human-robot shared task executions demonstrate, that flexible teaming can accelerate task execution [6]. The qualitative user evaluation of our human-robot communication app indicate, that users generally accept this mode of interaction despite a need for improvements regarding the user interface and communication timing [7].

IV. CONCLUSION AND FUTURE WORK

Teams composed of humans and robots will be an essential part of Factories of the Future. The FLEXCOBOT project investigates flexible teaming with dynamic task allocation and transitions between coexistence, cooperation, and collaboration. We have contributed a toolchain that supports end-users throughout all stages of system operation, from initial setup by calibration via task modelling to dynamic execution. This short paper complements our prior experiments with an initial user evaluation of the calibration step. Combining qualitative previous results with those regarding calibration we suggest, that the overall system may be operated by end-users without expert knowledge on robotics. We have moreover observed from a quantitative point of view, that calibration can me managed in less than 20 minutes, while composing complex task models takes no more than ten minutes. Thus, the span of time between having the system hardware installed and deploying the first shared task can be kept below 30 minutes. These results are of course limited by the application domain implemented in the prototype. We currently support pick&place operations with simple objects, and an assembly skill, where the robot holds a receiving part, while the human attaches mounting part. Although these skills cover typical applications conceptually, future work may target the extension and re-evaluation of the toolchain in the context of industrial use cases. Furthermore, questions during the calibration user evaluation indicate a need for a more intuitive indicator for LIDAR calibration precision. One may address this issue by providing a traffic light alike feature based on reprojection errors before launching large-scale evaluation of the overall system.

REFERENCES

[1] A. Perzylo, *et al.*, “SMErobotics: Smart Robots for Flexible Manufacturing,” *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 78–90, mar 2019.

[2] J. Kildal, *et al.*, “Potential users' key concerns and expectations for the adoption of cobots,” *Procedia CIRP*, vol. 72, pp. 21–26, 2018.

[3] M. Bender, *et al.*, “Lightweight robots in manual assembly – best to start simply!” Fraunhofer Institute for Industrial Engineering IAO, Stuttgart, Tech. Rep., 2016.

[4] G. Michalos, *et al.*, “A method for planning human robot shared tasks,” *CIRP Journal of Manufacturing Science and Technology*, vol. 22, pp. 76–90, aug 2018.

[5] L. Johannsmeier and S. Haddadin, “A Hierarchical Human-Robot Interaction-Planning Framework for Task Allocation in Collaborative Industrial Assembly Processes,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2017.

[6] D. Riedelbauch and D. Henrich, “Exploiting a Human-Aware World Model for Dynamic Task Allocation in Flexible Human-Robot Teams,” in *IEEE Intl. Conference on Robotics and Automation (ICRA)*, Montréal, 2019, pp. 6511–6517.

[7] D. Riedelbauch, S. Schweizer, and D. Henrich, “Skill Interaction Categories for Communication in Flexible Human-Robot Teams,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, to appear)*, Macau, 2019.

[8] F. Steinmetz, A. Wollschlager, and R. Weitschat, “RAZER - A HRI for Visual Task-Level Programming and Intuitive Skill Parameterization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1362–1369, 2018.

[9] C. Paxton, *et al.*, “CoSTAR: Instructing collaborative robots with behavior trees and vision,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 564–571.

[10] M. Kraft and M. Rickert, “How to teach your robot in 5 minutes: Applying UX paradigms to human-robot-interaction,” in *IEEE International Symposium on Robot and Human Interactive Communication*, Lisboa, 2017, pp. 942–949.

[11] D. Riedelbauch and D. Henrich, “Fast Graphical Task Modelling for Flexible Human-Robot Teaming,” in *50th International Symposium on Robotics (ISR)*, Munich, 2018, pp. 420–425.

[12] P. Rückert, *et al.*, “Calibration of a modular assembly system for personalized and adaptive human robot collaboration,” *Procedia CIRP*, vol. 76, pp. 199–204, 2018.

[13] T. Werner, D. Harrer, and D. Henrich, “Efficient, Precise, and Convenient Calibration of Multi-camera Systems by Robot Automation,” in *Intl. Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, Patras, 2018, pp. 669–677.

[14] K. Darvish, *et al.*, “Interleaved Online Task Planning, Simulation, Task Allocation and Motion Control for Flexible Human-Robot Cooperation,” in *27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Nanjing, 2018, pp. 58–65.

[15] N. Nikolaos, *et al.*, “On a shared human-robot task scheduling and online re-scheduling,” *Procedia CIRP*, vol. 78, pp. 237–242, 2018.

[16] M. Shah, R. D. Eastman, and T. Hong, “An overview of robot-sensor calibration methods for evaluation of perception systems,” in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM Press, 2012, pp. 15–20.

[17] K. Daniilidis, “Hand-Eye Calibration Using Dual Quaternions,” *International Journal of Robotics Research*, vol. 18, pp. 286—-298, 1998.

[18] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, 2004, pp. 2301–2306.

[19] R. H. Andersen, T. Solund, and J. Hallam, “Definition and initial case-based evaluation of hardware-independent robot skills for industrial robotic co-workers,” in *41st International Symposium on Robotics*, Munich, 2014, pp. 1–7.

[20] X. Niu, H. Ding, and Y. Xiong, “A hierarchical approach to generating precedence graphs for assembly planning,” *International Journal of Machine Tools and Manufacture*, vol. 43, no. 14, pp. 1473–1486, 2003.