Proceedings of the 27th International Conference on Automation & Computing, University of the West of England, Bristol, UK, 1-3 September 2022

From human-human collaboration to human-robot collaboration: automated generation of assembly task knowledge model

Yingchao You, Ze Ji, Xintong Yang, and Ying Liu
The School of Engineering, Cardiff University
corresponding author: Ze Ji; e-mail: jiz1@cardiff.ac.uk

***Abstract*—Task knowledge is essential for robots to proactively perform collaborative assembly tasks with a human partner. Representation of task knowledge, such as task graphs, robot skill libraries, are usually manually defined by human experts. In this paper, different from learning from demonstrations of a single agent, we propose a system that automatically constructs task knowledge models from dual-human demonstrations in the real environment. Firstly, we track and segment video demonstrations into sequences of action primitives. Secondly, a graph-based algorithm is proposed to extract structure information of a task from action sequences, with task graphs as output. Finally, action primitives, along with interactive information between agents, temporal constraints, are modelled into a structured semantic model. The proposed system is validated in an IKEA table assembly task experiment.**

Keywords-human robot collaboration; learning from demonstration; assembly; human centric manufacturing

I. INTRODUCTION

With the increasing demand for human-robot collaboration (HRC) in manufacturing scenarios, task-level planning systems were proposed to generate collaborative robot motions at different levels of abstraction [1] [2]. However, these systems typically require some form of prior knowledge about the task as prerequisites, such as task graphs models and grounding skills. In most existing works, such task knowledge is usually pre-programmed by domain experts [3]. Manually specifying the task graphs and action primitives by domain experts is time-consuming and not user friendly. Thus, it is highly desirable to enable robots to perform them automatically. This paper proposes a system that segments and interprets primitive actions of an assembly task from video demonstrations, constructs the task graph, builds semantic model of action primitive as robot skill library and transfers them to enable human-robot collaboration, based only on demonstrations of human-human collaboration.

Robot learning from demonstrations (LfD) has seen a fast development in recent years [4], especially in manipulation tasks. The methods and learning outcomes vary according to the content of the demonstration. The assembly demonstration is a structured activity that normally contains several subtasks and many action primitives. An efficient two-step solution has been proposed to extract task knowledge from such multi-step demonstrations [5], [6]. It first segments the demonstration into primitive actions using heuristic rules based on human knowledge and then represents the demonstrated behaviours using structured graph models. In this work, we adapt this method, using a vision-based parser to track the motions of the demonstrators and objects, defining a set of heuristic rules to segment the demonstration into primitive action sequences, and extracting task knowledge models from these segments.

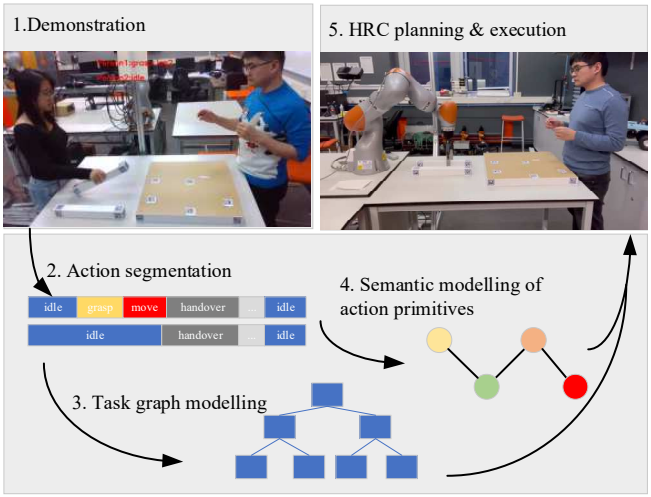

Figure 1. A system graph shows the procedures of assembly skills transferring from demonstrators to robots.

A task graph provides the task structure information required by the robot to plan for the task with uncertainties. One kind of task graph is the and/or graph [7], which is a widely used hierarchical model. An algorithm was proposed recently to generate and/or graphs automatically by recognizing the sequential and independent relationships of primitive actions [8]. However, this method assumes that "the sequential actions are not interrupted by parallel actions during the demonstration". This assumption is not always true. The reason is that the execution of parallel actions is independent. Thus, the sequential actions may be interrupted by parallel actions. To address situations when this assumption does not apply, we propose an algorithm that models the task structure into a directed graph by identifying action-action relationships. The generated directed graph is easily transformed into an and/or graph.

Different from LfD from a single agent, learning from dual agents is complex as agents may perform interactive actions in the demonstration, such as handover. Thus, action pair is present to describe interactive actions. In addition, the temporal constraints of an action pair are analyzed. The action primitives as well as their interdependencies are stored in a semantic model, which provides query and reasoning interfaces that are easy to use by the robot.

With the aim of transferring task knowledge from human-human collaborations to human-robot

collaborations, this work studies the automated generation of assembly task knowledge models. The key contributions of this work are summarized below:

- We propose a vision-based parser that is capable of real-time segment human-human demonstrations into sequences of action primitives without prior training.
- We provide an algorithm to automatically extract task structure knowledge and generate task graphs from action sequences.
- We construct a semantic model as a library of the learned skills, with interfaces for task planning in HRC.
- We design an experiment to validate the proposed methods. In the experiment, an IKEA table is assembled by two people in a collaborative way. Through observation, a Kuka iiwa LBR robot could learn to collaborate with a human in the task.

In the rest of the article, section II gives a brief overview of LfD and task knowledge modelling. Section III, components of the proposed system are provided in terms of a vision-based parser, task graph modelling and semantic models. Then, an assembly experiment is designed to test the proposed methods. Finally, our paper concludes in section V.

II. RELATED WORK

In the HRC content, knowledge engineering normally contains the experience acquisition, knowledge interpretation, constraints analysis, and modelling of task knowledge, with the task model being the output. There are various task model acquisition methods, including manual specification [8], [9], interactive learning, and learning from demonstrations. Anahita, etc. [10] proposed an interactive learning method where a human can teach a robot to construct hierarchical task models through demonstrations based on the structure information of objects and data flow between tasks. However, generating task knowledge models from demonstrations in HRC, which is the core of our method, has been paid limited attention until now.

Knowledge interpretation is one of the most important procedures of knowledge acquisition from a real-world demonstration. Despite the advancement in computer vision techniques, automatic acquisition of symbolic task representation of LfD-acquired skills remains difficult [2]. An effective solution of symbolic abstraction is to interpret the demonstration and segment actions by applying intuitive physical knowledge. In the literature [11], an ontology-based parsing method was used to reason daily activities in virtual reality (VR) environment, recognising basic actions such as take, reach, etc. The parser is based on a VR engine, which provides state information about agents and objects. In this work, a vision-based phaser is proposed to interpret the demonstration in the real world.

The generation of a hierarchical task model essentially is a process of interpreting the relationship of action primitives. In terms of the learning methods, Hayes *et al* [2] provided a transformation algorithm from task graph to hierarchical task model. Cheng *et al* [8] proposed a sequential/parallel task model and a corresponding algorithm that can identify the relationship of primitives. However, these works did not integrate with LfD based knowledge interpretation methods. We propose a novel task graph generation algorithm that integrates with the LfD based parser module.

When learning from complex activities, an effective way is that a structured model can be constructed to store the interpreting knowledge of the demonstration. In [12], the obtained semantic information transformed into an ontology-based model, know-rob [13], and it provides interfaces for querying and reasoning for action planning.

III. SYSTEM

All components of the proposed system are shown in Figure 1. which consists of five steps. In step 1, two demonstrators conduct an assembly task collaboratively. Then, a vision-based parser is used with a set of rules to interpret the demonstration into sequences of action primitives. In step 3, through analysing action relationships, task structure information is extracted and then a task graph is constructed. Step 4 identifies the semantic information of the grounding skills and store this knowledge into a semantic model. The task graph and semantic model are used for symbolic-level planning in an HRC assembly task.

Demonstrations are performed in a real lab environment, and two demonstrators conduct assembly tasks in a master-slave way. One of the partners is the principal operator, and the other one performs as an assistant. The ultimate goal of the robot is learning to act as an assistant to humans in assembly activities.

A. Vision-based parser

The scheme of the parser is shown in Figure 2. In general, firstly the skeleton model of demonstrators and the simplified model of objects are modelled based on the visual and depth information, and the 3D position info is obtained. Then, the kinematic information of both human and objects are calculated. The human poses are recognized based on the obtained status of humans and objects in real-time.

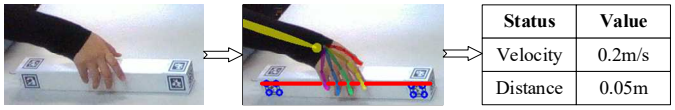

Figure 2. The video demonstration is interpreted. The skeleton model of humans is tracked and constructed, and the table leg is abstracted as a line segment. The velocity of the human wrist point is detected and the distance between the table leg and the hand is calculated.

Thanks to the quick development of computer vision techniques [14], tracking of humans and objects are realized in real-time. Human poses in each frame are transformed into a 3D skeleton model, which contains the

TABLE I. THE STATE VARIABLES(*SV*) OF HANDS OF DEMONSTRATORS AND OBJECTS

sv	Types	Examples	Description
handmoving (*h*)	boolean	handmoving (hl_lefthand) = true	The velocity of moving hand is above 0.15m/s
inhand(*h*)	objects	inhand (hl_lefthand) = leg1	The finger of the hand is attached around an object and finger-hand distance < 5cm
hand2hand (h_1,h_2)	boolean	hand2hand (h_1_lefthand, h_2_righthand) = true	The distance between hand1 to hand2 < 15cm, and at least one of hands has an object inhand,
intouch (o_1,o_2)	boolean	intouch (leg1, table) = true	The minimum distance between object1 and object 2 is less than 3cm.

Note: *h* denotes the hands of humans; *o* denotes the object s

ACTION CLASSIFICATION RULES

Action / *sv*	idle	grasp	move	handover	screw	hold
handmoving (*h*)	-	F	T	-	-	F
inhand(*h*)	Ø	¬Ø	¬Ø	¬Ø	¬Ø	*b*
hand2hand (h_1, h_2)	F	F	F	T	F	F
intouch (*l*, *f*)	-	T	F	-	F	-
intouch (*l*, *b*)	-	-	F	-	T	-

Note *l*: to be assembled objects in the experiment, table legs; *f*: experimental platform; *b:* to be assembled objects in the experiment, tabletop. T: true; F: false; -: Variables that do not affect classification results.

cartesian coordinates of key points of humans. Objects are marked by Apriltage [15] to get the key point coordinates of objects in cartesian space, which can be easily transformed to the pose and position information.

Due assembly tasks are mainly finished by the hands of workers, the velocity detection of hands is necessary. Given the skeleton model of demonstrators, the speed v_t of the hand at time *t* is derived from the position of the hand at different points in time. The velocity of speed can be approximated by wrist joint speed. Thus, v_t is formulated as follows:

$$v_{t_i} \approx dis\left(w_{t_i}, w_{t_j}\right) fps/(t_i - t_j)$$

v_t represents the speed of the hand; w_t is the cartesian coordinates of the waist at time *t*; *fps* denotes frame rate per second; *dis()* denotes a function that returns the distance between two points.

Object detection or tracking is required to obtain the poses of objects in each task and can be simplified using methods such as colour-based detection [16]. In this work, the spatial position of objects in each frame is abstracted into basic geometrical elements, such as line segments, flat surfaces. For example, a table leg, which is a cuboid, is simplified as a line segment l_{p_1,p_2}, and p_1, p_2 are endpoints of the line segment (refer to Figure 2.). The distance from the hand to the table leg is represented by the minimum distance between key points of the hand to the line segment.

$$d_t = min(dis(l_{p_1,p_2}, h_t))$$

h denotes the key points of hand at time *t; min()* is a function that returns the minimum value in the matrix.

Due to the complexity of the assembly process, it is difficult to recognize action segment activities directly from the demonstration. Thus, a state-of-the-art method [6] is adapted and extended in this work. This is a knowledge-based method, which applies intuitive physical knowledge to interpret the demonstration. The segmentation process consists of two steps: (1) defining state variables; and (2) classifying actions based on rules. The defined state variables are listed in TABLE I. in terms of types, examples, and description, the state variables consist of two types, hand state variables, and environment state variables. handmoving(*h*), in-hand(*h*) and hand2hand(h_1, h_2) belongs to previous category. Note hand2hand(h_1, h_2) is extended that is set to monitor the interaction between different agents. In addition, intouch(o_1,o_2) is set to monitor the interaction of objects. The physical knowledge of action is designed as rules to classify the action, and it is listed in 0Handover is a critical action between different agents, where one agent passes objects to another one. This action starts when the hands of the two agents are approaching each other and one of the hands is grasping an object. The finish point is that the object pass to the other agent and hands are gradually far away from each other. The segmentation points are set when the action status changes, and action segmentation is realized. The segmentation information is used to automatically generate task graphs and semantic models.

B. Automated task graph construction

By applying the action segmentation, the approach in Section 3.A. the action sequences $\Xi = [\xi_1, \xi_1, \ldots \xi_n]$ of demonstrators is extracted from the performed demonstration, where *n* is the total number of demonstrations of assembly activity. $\xi_i = \left[a_1^i, a_2^i, \ldots a_{m^i}^i\right]$ means sequences of action primitives with m^i action units $a_{m^i}^i = [motion, objects]$ that normally contains a motion and a relevant object, for example, $a_{m^i}^i = [move, table\ leg1]$. In an assembly task, the actions in different demonstrations are usually the same, but the sequence of primitives vary. Besides, the action sequences may contain actions unrelated to the task, such as idle states. These actions should be removed from action sequences.

The proposed method aims to construct a task graph from action sequences Ξ, which is easy to transform into a hierarchical task model. The key of constructing task graph $g = [n, e]$ is to identify the relationship of all action primitives, with node n representing the action primitives and edge e denoting the transition between actions. Based on the identified relationships, it is easy to connect the primitives to form a task graph. We define a series of relationships of action primitives and the corresponding identification methods. Then, an algorithm is proposed that forms a task model. The identification method of nodes and edges in the task graph is introduced based on the relationship of primitives.

Algorithm 1: Task graph generation

```
      Input Ξ
      Output graph
1:    init graph, A, headnodeSet, endnodeSet
2:    A = generationPRM (Ξ)
3:    initAction = findInit Action (A)
4:    headnodeSet = initAction
5:    while then do
6:      for each action a1 in headnodeSet do
7:        actions= find FollowupAction (a1)
8:        endnodeSet append (action)
9:        for each action a2 in action do
10:         graph.addEdge (a1,a2)
11:       end for
12:     end for
13:     If endnodeSet is empty then
14:       Break
15:     Else then
16:       headnodeSet = endnodeSet
17:       endnodeSet.clear
18:     End if
19:   End while
```

In the industry assembly process, actions may have a dependent relationship with each other due to the physical features of products. Specifically, we define five relationships of action primitives, including pre-order action, post-order action, independent relationship, immediate predecessor action (IPA), immediate successor action (ISA). First of all, we define the primitive relationship matrix (PRM), denoted by $\boldsymbol{A}$ in the equation below, for a task, which represents the specific relationship of different primitive. It is defined as:

$$A = \begin{bmatrix} \alpha_{a_1 a_1} & \cdots & \alpha_{a_n a_1} \\ \vdots & \ddots & \vdots \\ \alpha_{a_1 a_n} & \cdots & \alpha_{a_n a_n} \end{bmatrix}$$

When $\alpha_{a_i a_j}$=1, action a_i is the **preorder action** of a_j. The a_j should be finished before a_i starts. The pre-order action produced consequence is required by the execution of action a_i. The preorder action set of a_i is represented as Φ_i. To identify that a_j is the pre-order action of a_i, all action sequences Ξ are going through to check the occurrence of a_i, a_j. If a_j occurs before the occurrence of a_i in all sequences, a_j is the pre-order action of a_i; otherwise, not.

When $\alpha_{a_i a_j}$= -1, action a_i is the **post order action** a_j. Action a_i should be finished before a_j starts. The relationship of post order is the inverse of preorder. The identification process is similar to it.

When $\alpha_{a_i a_j}$= 0, action a_i, a_j are **independent**. The action a_i, a_j can occur in any order. If the relationship of a_i a_j is not post order or pre order, a_i, a_j are independent.

If a_i is **ISA** of a_j, the occurrence of a_i is directly after action a_j. The mathematic formulation of ISA is:

$$\exists i \in (1, n)\,, \Phi_i = \Phi_j + a_j$$

If a_i is the **IPA** of a_j, the occurrence of a_j is directly before action a_i. ISA is the inverse of IPA. If a_i is the IPA of a_j, a_j is the ISA of a_i.

The proposed algorithm is shown in Algorithm1. The input of the algorithm is Ξ, the sequence of action primitives extracted from the demonstration. The output Graph is a task graph, which shows the structure information of an assembly task. Headnodeset is used to store a series of action primitives, which is regarded as the head node of an edge in the graph. Endnodeset is a list that contains the corresponding endnode of headnodeset. Line 1 initializes the variables Graph, $\boldsymbol{A}$, headnodeset, endnodeset to be empty. Line 2 generates a PRM matrix with Ξ as input by using the identification method in Section 3.B. Line 2 find the initial nodes of the Graph, by using the following formulation.

$$\exists i \in (1, n)\,, \Phi_i = \phi$$

When the preorder action set of an action primitive is empty, the action is regarded as the first node of Graph. In line 4, the headnodeset is assigned with the values of InitialAction. Line 5-19 find the endnode of headnodes and then construct edges in graph till to end iteratively. Line 6-8 find the endnodeset of each action in headnodeset. The searching rules of endnodeset is based on the ISA identification rule. Line 9-11 add edges in Graph by connecting headnotes and endnodes. Line13-15 define the break rule: the graph is not updated in this loop. When endnodeset is empty, the graph does not add a new edge in this loop, then break the loop. Otherwise, endnodeset is used as a new headnodeset, in the meantime, the old headnodeset is cleared.

C. Structured, semantic model

As the vision-based parser discover sequences of the action of two demonstrators, segmented grounding skills are modelled into a semantic model in this section. During the task execution, workers conduct some collaborative actions. While modelling their collaborative behaviours, temporal constraints analysis of actions is necessary.

The information stored by the semantic model is summarized as follow: (1) properties of the action primitives; (2) the manipulated objects information; (3) the action constraints of the interaction between demonstrators.

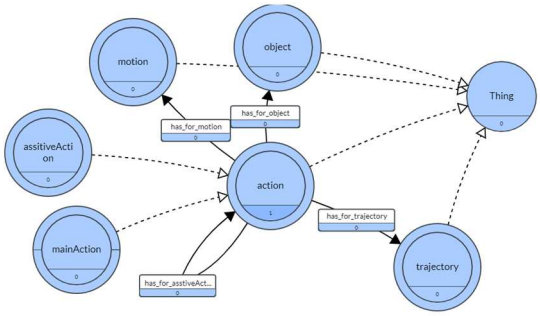

Figure 3. The semantic model of action primitives

1) Constraint analysis

The learned action primitives can be categorised into the main action $\boldsymbol{a}_m$ and assistive actions $\boldsymbol{a}_a$, performed by the principal operator and the assistant respectively. In an

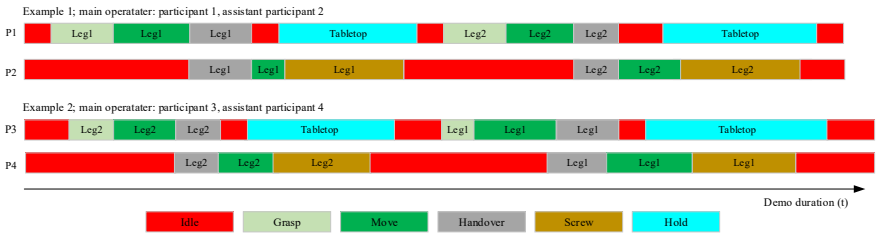

Figure 4. Two examples of action segmentation outcome in assembly experiment

assembly task, the cooperative behaviours of the operators generally consist of active and assistive actions. Thus, we construct cooperative action pair $p = [\boldsymbol{a}_m, \boldsymbol{a}_a]$. The identified rule of p is that: if $\boldsymbol{a}_m$ and $\boldsymbol{a}_a$ work on an object physical interactively, the $[\boldsymbol{a}_m, \boldsymbol{a}_a]$ is action pair. Especially, actions in p is not necessarily a single action unit, but sometimes a sequence of actions.

The temporal constraints exist between actions in p. The execution time of action in the demonstration is an interval $[t_a^l, t_a^r]$, and t_a^l is the beginning time and t_a^r is the end time. We define two types of temporal constraints, (1) **prior**. The assistive action should be done before the main action. The constraint is formulated as follows:

$$t_{a_a}^l < t_{a_a}^r \leq t_{a_m}^r,$$

(2) **meantime**. The assistive action should be done during the execution of the main action. The constraint is formulated as follows:

$$t_{a_a}^l \leq t_{a_m}^r < t_{a_m}^r \leq t_{a_a}^r$$

2) Semantic model

The constructed semantic model is visualized in Figure 3. The properties of actions have motion, objects and trajectories. The motion is classified into mainAction and assistiveAction. The semantic model contains action pairs that contain the joint action of human and robot. During the HRC execution, we utilize the semantic model to control the robot, the robots can do joint action with humans based on the observation. The model can record action primitives and constraints. The model provides interfaces for querying and reasoning, and it can be used as a robot skill library.

IV. EXPERIMENT

In order to evaluate the proposed methods, a real assembly task is set by using an IKEA table (LACK) that has a tabletop and four table legs. We simplify the assembly process, with only two legs to be screwed into the tabletop, as assembling the other two legs is repetitive work. The objects to be assembled are placed on a platform. Two participants stand on the opposite sides of the platform. The main operator stands near the tabletop and far away from the table legs. Thus, the assistant is asked to handover the legs to the main operator. Besides, the assistant is asked to hold the tabletop to keep it stable, while the main operator is screwing the legs. All participants have read the assemble instructions of the table. 4 individuals participated in performing the assembly task. In total, the system records and interprets the assembly process 18 times. The demonstrations are recorded by an Intel RealSense D435 camera.

A. The performance of action segmentation

Figure 4. displays examples of the action segmentation of the human-human collaboration demonstration in the assembly task, including the duration of action primitives and the transition between different actions. The action primitives performed by different demonstrators were mostly the same, with some differences in duration and order.

In order to evaluate the accuracy of the action recognition of the proposed methods, we obtained the ground truths by playing back the recorded videos and manually labelling the actions of every video frame. If the value is different from the result identified by the algorithm, the recognition result is incorrect. Accordingly, the recognition accuracy of all 18 demonstrations was 91%. The main reason for the failures is the misestimation of visual tracking. For example, hands may be blocked by objects (e.g., legs), which leads to the inaccurate position estimation of the captured key points of hands.

B. Task graph generation

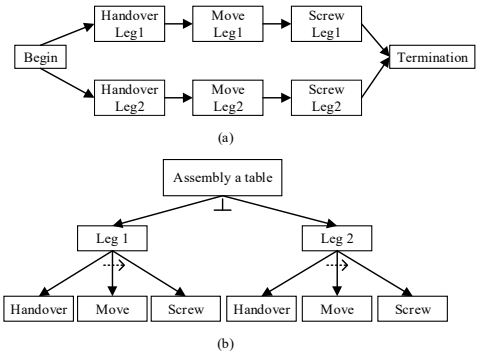

Figure 5. The generated task graph and transformed hierarchical task model

The input of the task generation algorithm in this task only needs the action sequences of two demonstrations (Figure 4.) First of all, all 'idle' actions that are unrelated to the task are eliminated. Then, the segmentation data is fed into algorithm 1 to generate the task graph, shown in Figure 5. By using the algorithm in [2], a hierarchical task model is obtained by transforming the task graph (Fig. 5b).

The number of required demonstrations to completely generate a task graph using the proposed algorithm depends on the structure of the task. Suppose a task has q levels in hierarchical model and $p \in [1,2,\dots,q]$ level has i_p independent nodes, which has c_{i_p} children. $max\left(c_{i_p}\right)$

means the number of the children of the node with the maximum children in level p. The number of required demonstrations to construct a complete task graph is $\prod_p^q max\left(c_{i_p}\right)$ by using the proposed algorithm. The time complexity of our algorithm is $O(m^2n^2)$, where m is the number of demonstrations and n is the number of action primitives.

C. Semantic model

In symbolic-level HRC task execution, the semantic model can be used to search for action primitives as well as the assistive action based on the main operator's action. In this section, we present two examples of how these can be done. Besides, the invoking mode and functions of this model during HRC execution are illustrated.

The first function of the model is to search for an action primitive. The search requires a motion name and an object name to be given. An example of the search result is shown as follows.

action: (name: 'hold_table_top', type: 'assitive action'
motion: [name: 'hold'],
object: [name: 'table_top'],
trajectory: [list])

Another function is to search for a corresponding assistant action. The search requires the main operator's action name and is based on the rules of assistive action. The example shown below queries the assistive action of action 'screw_table_leg1'.

action: (name: 'screw_table_leg1', type: 'main action'
motion: [name: 'screw'],
object: [name: 'table_leg1'],
trajectory: [list],
assistiveType: [type: 'meantime'],
assistiveAction: [action: 'hold_table_top'])

D. Assembly task execution

We test our constructed models in an assembly experiment using a Kuka iiwa LBR robot. A task graph-based action planner is used [3]. We modify the planner by replacing the Bayesian model with our proposed semantic model for querying the assistive action. Figure 6. shows that a robot is assisting a person in an assembly task, where the person is assembling an IKEA table. The action of the robot is planned by the planner. An experimental video is attached to this paper.

Figure 6. A robot collaborates with a person in an assembly task.

V. Discussion and Conclusion

Preparing knowledge models for symbolic planners in HRC is a time-consuming, user-unfriendly task. In this letter, we presented a system for automated knowledge model generation through visual demonstration interpretation, task graph modelling and semantic model generation. Despite the complexity of the assembly task, the parser achieved an accuracy of 91%. The task graph model was generated using only two demonstrations with an acceptable time complexity $O(m^2n^2)$. The semantic model was tested in a real-world assembly experiment using an IKEA table.

Acknowledgment

Yingchao You, Xintong Yang thank the Chinese Scholarship Council (CSC) for providing the living stipend for their Ph.D. programmes (No. 202006020046, No. 201908440400).

References

[1] V. Shivashankar, K. N. Kaipa, D. S. Nau, and S. K. Gupta, 'Towards integrating hierarchical goal networks and motion planners to support planning for human robot collaboration in assembly cells', 2014.

[2] B. Hayes and B. Scassellati, 'Autonomously Constructing Hierarchical Task Networks for Planning and Human-Robot Collaboration', 2016 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA, pp. 5469–5476, 2016.

[3] F. Stramandinoli, A. Roncone, O. Mangin, F. Nori, and B. Scassellati, 'An Affordance-based Action Planner for On-line and Concurrent Human-Robot Collaborative Assembly', 2019.

[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, 'Recent advances in robot learning from demonstration', Annu. Rev. Control Robot. Auton. Syst., vol. 3, pp. 297–330, 2020.

[5] M. Diehl, C. Paxton, and K. Ramirez-Amaro, 'Automated Generation of Robotic Planning Domains from Observations', ArXiv Prepr. ArXiv210513604, 2021.

[6] K. Ramirez-Amaro, M. Beetz, and G. Cheng, 'Transferring skills to humanoid robots by extracting semantic representations from observations of human activities', Artif. Intell., vol. 247, pp. 95–118, 2017.

[7] R. A. Knepper, D. Ahuja, G. Lalonde, and D. Rus, 'Distributed assembly with and/or graphs', 2014.

[8] Y. Cheng, L. Sun, and M. Tomizuka, 'Human-Aware Robot Task Planning Based on a Hierarchical Task Model', IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1136–1143, 2021.

[9] V. Montreuil, A. Clodic, M. Ransan, and R. Alami, 'Planning human centered robot activities', in 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 2618–2623.

[10] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, 'Interactive hierarchical task learning from a single demonstration', in Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, 2015, pp. 205–212.

[11] T. Bates, K. Ramirez-Amaro, T. Inamura, and G. Cheng, 'On-line simultaneous learning and recognition of everyday activities from virtual reality performances', in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 3510–3515.

[12] A. Haidu and M. Beetz, 'Automated acquisition of structured, semantic models of manipulation activities from human VR demonstration', in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 9460–9466.

[13] M. Tenorth and M. Beetz, 'KnowRob: A knowledge processing infrastructure for cognition-enabled robots', Int. J. Robot. Res., vol. 32, no. 5, pp. 566–590, 2013.

[14] Z. Cao, G. H. Martinez, T. Simon, S. Wei, and Y. A. Sheikh, 'OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields', IEEE Trans. Pattern Anal. Mach. Intell., 2019.

[15] E. Olson, 'AprilTag: A robust and flexible visual fiducial system', in 2011 IEEE international conference on robotics and automation, 2011, pp. 3400–3407.

[16] K. Ramirez-Amaro, M. Beetz, and G. Cheng, 'Automatic segmentation and recognition of human activities from observation based on semantic reasoning', in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 5043–5048.