A "Hybrid" Personalized Model for Collaborative Human-Robot Object Manipulation

Maren Roettenbacher[1] and Andreas Riener[2]

***Abstract*— This work proposes a hybrid Markov Decision Process (MDP) based approach for planning and decision-making in finite horizon, complex collaborative human-robot object manipulation tasks. The approach is hybrid in the sense that the full model state-space is defined by an object-centered rule-base, while the model parameters are trained using an apprenticeship learning approach, i.e., observing humans performing the tasks.**

Current research focus is on household scenarios that are characterized by multiple alternating but recurring users and tasks. The system is tailored to fit the specific requirements as well as the limitations resulting from the chosen domain, namely unskilled trainers and limited amount of data samples with the core goals of easy system reconfiguration, continuous personal adaptation, task fluency, reliability and traceability of robot decisions.

I. INTRODUCTION

Robots are increasingly acting as collaborators in social and industrial activities. With emerging dexterity and skill set of robots, the range and flexibility of task sharing will further evolve. Thus, it is a good hypothesis that human expectations on robot task performance will increasingly resemble those upon other humans they work with: robots should assist without imposing additional cognitive workload to the human, while achieving a fluent task execution.

One major aspect is an anticipatory, reliable/robust (and safe) robot behavior, i.e., an implicit understanding of human intentions and of the common goal - based on the current context (prior knowledge and constraints) and non-verbal human cues and actions [1][2]. In a long-term or recurring interaction, continuous adaptation to individual manners and preferences based on implicit (e.g., facial expression) and explicit (e.g., utterances) feedback mechanisms and a growing sample base is required.

A. Problem Setting

The focus of this work is on collaborative human-robot manipulation scenarios with an alternating/interdependent blended action structure or more colloquially speaking: assisting in complex everyday tasks, where a "helping hand" is useful or required. For the scope of this research, collaboration takes place in a household setting with the average (technologically unskilled) customers or co-workers (note: we expect that the approach is transferable to other domains e.g., industrial assembly, where the addressed type of task might be even more prominent). Collaboration is currently limited to one collaborator at a time. However, preferences of each individual should be respected by the robot. The tasks to achieve have finite horizon and can be seen as subsets of a more complex long-term planning system in order to achieve a tractable hierarchical reasoning structure. The proposed planner does not work on the trajectory or control level but at the semantically higher level of the task network.

For the purpose of reasonable planning and decision making, human and robot need a shared (mental) model of the task, meaning that they have a similar understanding of the situation, leading to similar decisions. The specified interaction type induces a strong object focus of the addressed task set. Hence, the common ground is mainly (clearly not entirely) defined by the consistent knowledge about the effects of actions upon and amongst objects.

Consistent knowledge about objects and task constraints can be formulated in an ontology or rule base, respectively. From the ontology, the possible (parametric) state-space of a task model can be enumerated [3]. Rules can limit the parameter range of the model or certain states (i.e., deterministic action choice, task order, etc.).

The basic model needs to be configured for the specific user/scenario combination. Usually, the collaborating persons do not have the necessary capabilities to "program" the robot for a complex interaction task. Moreover, this would be a very tedious and error prone process as person-specific knowledge is not deterministic and not always directly accessible (yet sometimes irrational). There are not only inter-person differences but also intra-person variations within the task.

However, the missing belief about the human action/reaction structure is implicitly available in the human and can be drawn from the observation of human task execution ("expert" demonstrations). This fact has been proven by research on apprenticeship learning [4]. [5] captures the interaction dynamics in a mixed-observability Markov Decision Process (MOMDP) for human-robot collaboration. This is a very appealing approach for learning the characteristics of a user centric task model (especially as long-term learning approach in combination with reinforcement learning methods) that are almost impossible to model in a generally valid fashion. For this "on-site configuration", our approach is based on the understanding of non-verbal actions. Utterances and further explicit and implicit signals might not be neglected but are out of scope (a POMDP

[1]Maren Roettenbacher is with the Faculty of Electrical Engineering and Computer Science, Technische Hochschule Ingolstadt and with the Institute for Pervasive Computing, JKU Linz, 4040 Linz, Austria maren.roettenbacher@thi.de

[2]Andreas Riener is with the Faculty of Electrical Engineering and Computer Science, Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany andreas.riener@thi.de

for multimodal human-robot interaction trained from human-human demonstration can be found in [6]).

B. Domain-specific Constraints and Design Choices

If we consider a "real world" setting, there are some limitations that have to be respected when defining the model generation process.

- The trainers, as opposed to the above specification of apprenticeship learning, are no "real" experts. They act to the best of their knowledge (i.e., wrong instantiations might occur). In addition to the unintentionally wrong samples, people might try to intentionally feed the robot with wrong behavior.
- There will be only a limited amount of training data available.

These factors impact not only task configuration but especially the potential for learning reliable object attributes and action effects. This is practically not possible. Therefore, it is proposed that object (class) attributes are trained a priori, e.g., in specialized facilities in a supervised fashion and used for efficient model generation.

This predefinition of verified and consistent object knowledge has another very practical implication considering the "real world" domain. It addresses the problem of verification, traceability of decisions and liability of the vendor, if the robot causes any physical or economic damage (clean the Teflon pan with steel wool because estimated reward is still higher than not completing the task at all). It prevents the robot from making fatal decisions due to wrongly learned or estimated model parameters in the reward based approach or due to the lack of completely well-defined goal states ("clean the Teflon pan from the dirt" instead of "clean the Teflon pan from the dirt without scratching the coating"). It assures that the robot will not "believe" wrong instantiations of object usage in a workflow, whether intentional or unintentional.

For the same reasons, it might be favorable to have deterministic rules on task interdependencies/constraints for (safety) critical sections of the workflow.

II. ILLUSTRATIVE EXAMPLE

Let us illustrate the scope of this work with a simple example. Imagine a table with a knife, two different glasses, a bottle, a bottle opener and a bread. It is easy to see that there is a functional relation between the glass, the bottle and the bottle opener as the functionalities are consistent with respect to tasks already performed by almost every human. As well, there is a relation between bread and knife. There might be a relation between knife and bottle known to humans, if the knife is used for opening the bottle, which represents a misuse of the object and should be ignored by the robot. So by simply knowing that the glass and the bottle can be either used for pouring something in or out and the bottle opener has the single functionality of opening a closed bottle while the bread and knife can be neglected when one of the aforementioned objects is used, the set of potentially rewarding decisions within this task is already limited as compared to the strict exploration approach.

However, it is not entirely predefined what the workflow looks like in a specific setting. For example, the human might prefer to have a certain drink in a certain type of glass and whishes to have ice in the drink, which is stored in the freezer compartment of the fridge. The robot has learned this while staying in the specific location simply by observing the humans performing the same tasks. Likewise, the robot might have learned how long it takes the human to open a bottle and thus is able to hand the glass in time without an explicit demand. The robot knows that there is a constraint between the duration of handing the glass and the duration of pouring something into the glass that should not be violated.

In the case of pouring something into a glass, robot failure causes some (more or less acceptable) annoyance to the human. In a different workflow like e.g. mounting a shelf together, the violation of the constraint between securing the object on the wall and the collaborator having finished the task of fixing it to the wall might cause severe injuries and the temporal action constraint should be specified in a deterministic way.

III. RESEARCH QUESTION

The question of how to capture user dependent task specifics and environment specifics gets especially relevant, when the system is used in a rather complex collaborative manipulation like e.g. in strongly intertwined and interdependent assembly tasks (for conciseness, we speak of collaboration as the closest form of interaction, implying direct interaction in working towards a common goal, using the same limited set of resources and having strong and immediate interdependencies between the actors' actions). Thus, we are specifically interested in providing and evaluating a methodology for planning and decision making in this workflow type. The chosen example workflows should reflect certain aspects, for which we want to evaluate whether they can be covered by our "hybrid" modeling approach:

- Have a task order that is not deterministic
- Have uncertain action effects of human actions
- Have a hidden object (e.g. in drawer, under lid)
- Have a safety critical constraint between two actions (e.g. heavy weight will fall)
- Have a property changing action (e.g. from "movable" to "not movable" or vice versa)
- Have an item with "separating" or "merging" character e.g. knife and bread, glue and paper, lock and key

Due to the lack of a life-size humanoid robot, we have to somehow abstract the example workflow either by using simplified and easy to handle objects and evaluate with a smaller robot or by shifting the experimental setup to some Augmented Reality (AR) environment, where the human handles real objects, while the virtual robot handles virtual objects. The AR approach might produce a more realistic sensation but requires the absence of actual physical interaction between objects within the workflows. In conjunction with the functional evaluation of the model, we are intending to conduct user studies to answer important human centric questions like:

- Does apprenticeship learning alone (before adaptation via reinforcement learning) provide an "initial guess" perceived as adequate - which amount of training data is required?
- Can we achieve improved user comfort by raising task fluency?
- Do the implicit reasoning capabilities perform well with respect to user comfort/trust?

IV. PROPOSED SYSTEM SETUP

The goal of this work is to define which and how object knowledge and task constraints can be formalized in a rule-base/ontology and how this knowledge can be used for the creation of a discrete MOMDP state-space. For action recognition, a generic classifier for movement primitives (characteristic approach-act-retract fashion motion) based on skeleton and object track data and computationally efficient relative features (e.g., distance, speed etc.) is developed. The classifier is applied to human task execution for apprenticeship learning. Figure 1 gives an overview of system components and interplay. As a further benefit, primitives can directly be mapped to the robot skill set on the semantic level and parameterized from the observation. This high level task can consequently be executed by the robot motion planner and control. In future work, the model can be optimized using reinforcement learning based on social signals (e.g., facial expression combined with body posture), utterances or performance metrics (robot or human idle time, time to completion, etc.) in the sense of a lifelong learning approach.

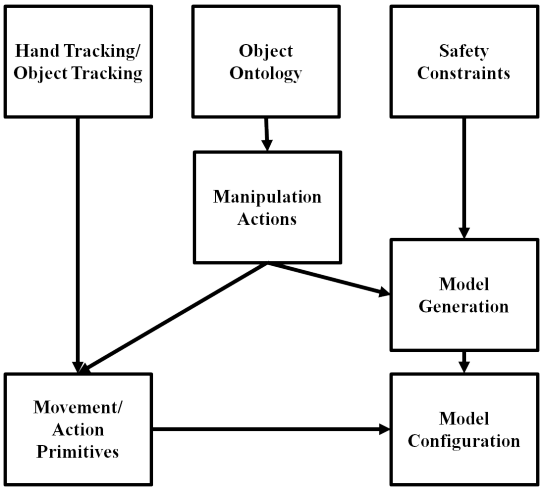

Fig. 1. System Overview

A. Mixed-observability Markov Decision Processes

A Partially Observable MDP is defined as a tuple $\langle S, A, O, T, Z, R, \gamma \rangle$ where S is the set of states of the world, A is the set of actions an agent can execute, O is the set of observations an agent can perceive, T is the probabilistic transition function $T(s, a, s') = P(s'|s, a)$ which gives the probability of ending in state s after executing action a in state s, Z is the probabilistic observation function $Z(s', a, o) = P(o|s', a)$ which gives the probability of receiving observation o when state s is reached via action a, R is the set of expected rewards $r(s, a)$ for each state-action pair and γ is the discount factor that specifies how much future rewards will be discounted. At each time-step, the machine is in some unobserved state $s \in S$. The machine selects an action $a \in A$, receives a reward $r(s, a)$ and transitions to (unobserved) state $s' \in S$, where s' depends only on s and a. The machine receives an observation o which is dependent on s' and a. [7]. Extending the concept of POMDPs with complete observability leads to Mixed-observability Markov Decision Processes (MOMDPs), where some state variables can be directly observed [8].

B. Object Ontology

In robotics, there is a wide range of problems to solve, before an actual flexible collaboration can take place. One key component for manipulation tasks is perception. The robot must possess of a generic object recognition capability in order to recognize unseen entities of an object class. It must not only possess of the physical ability of grasping the object (anthropomorphic gripper, payload, etc.) but also know how to grasp it in a stable way, without damaging it (sensitivity) and know about further handling constraints (do not handle full cup upside down or tilted) important to motion planning and control. Moreover, the robot has to be aware of the object affordances, meaning here: what is the intended object use and which effects will this induce. Most of these topics are still subject to ongoing research. However, for the purpose of this work, it is assumed that the robot is able to recognize objects, can flexibly grasp them and has substantial knowledge about limitations to object handling as well as about affordances.

For task model generation, we address the concept of affordances by specifying functional object properties. This step is at the current point solely based on human background knowledge and treated as generally valid. Likewise, the information could e.g. be retrieved from the actual (size, material) and perceived (which utility the object "indicates") properties of an object [9] in conjunction with the recognition step [10]. In this sense, we furthermore assume that perceived affordances are linked to real affordances ("pushable" button has a related functionality) and that affordances are independent of the actors' handling capabilities.

The specification of functional object properties can be done on a class level and be inherited within the class hierarchy or in a more specialized fashion on the entity level (e.g. *container can be used for content* ← *cup can be used for liquid* ← *coffee cup can be used for coffee*). Using more prior knowledge and specialized object descriptions renders the model less flexible but smaller and more reliable.

As a starting point, we define the attributes listed in Table I in order to characterize an object and find the reduction of the state-space.

The attribute description are deliberately kept at a semantically high level (e.g. use/use with as a placeholder for actual affordances) as they are intended for demonstrating

Attribute	Values	Assignment
Tangibility	Tangible, intangible	Mutually exclusive
Portability	Movable,static	Mutually exclusive
Structural Unity	Entity, separable	Mutually exclusive
Topological relations	Container, Multi-Container, content, none	Not exclusive
Utility	Usable, usable with, none	Not exclusive

TABLE I

SUGGESTION FOR ATTRIBUTE SET OF FUNCTIONAL OBJECT PROPERTIES

state space construction rather than actual action execution. A more complete definition of object handling ontology can be found in [12].

The attributes are chosen in order to define which primitive actions can be performed upon the objects, which interactions are viable, whether the objects can form new "'objects" when combined (topological state combinations like "'pen in box" or actual merging actions) in order to span up the potential MOMDP state space. When objects are combined (*water + cup = water in cup*), the *combination object* "'inherits" prominent properties of one of the objects. The attributes and the effects of combining objects are explained below:

- Tangibility: whether an object can be touched. Does not capture, whether an object cannot be touched because it is inside another object with a lid. Objects that require a container are treated as intangible (e.g. handling of fluids). Logical combination (OR) of tangibility property of combined objects (*water = intangible, water in cup = tangible*)
- Portability: whether an object can be changed in place. Does not capture whether the object cannot be moved due to the context e.g. if the object has been glued or nailed to something. Logical combination (AND) of portability properties of combined objects
- Structural unity: whether an object can be separated. Captures whether an object can have several values in same state (e.g. sugar can be on spoon AND/OR in sugar dispenser whereas cup can be in cupboard XOR on table). This property can vary in a context where related objects with splitting or merging property are present (knife, screwdriver, glue)
- Topological relation: whether an object is container or content. Objects can be one, both or none. This value also specifies whether a container can contain multiple objects at the same time (multi-container) and defines the possible object-in-object nesting depth of the state space.
- Utility (relation): whether an object is for standalone use or can be used in conjunction with another object. Objects can be one, both or none. This is an abstracted representation of further obligatory knowledge about object utility that is used for simplicity reasons when specifying the example object properties. Needs to describe object state variations due to action effects (*on/off, open/closed*)

As part of our research, we will try to find a formal notation of the above characteristics and enumerate a usable base MOMDP for several sample scenarios. We will iteratively refine the description methodology, unify the taxonomy over the sample scenarios and try to gain insights on the necessary depth of object descriptions.

C. Constraint Set

We are seeking for model adaptations representing spatio-temporal variability and temporal relations between parameterized actions (like e.g. achievable with Interaction Probabilistic Movement Primitives [13] on the trajectory level or by using discrete actions with continous parameter vectors [14] which permits reinforcement learning on parameter level with reward based on task performance metrics or human factors like task engagement [11]).

In regular workflows, parallel actions will occur, as an action is mostly extended over some period of time. This means, an action might start/end after a certain time span preceding/following the coworker action. As there is no explicit observation to trigger the state change, the duration since the last action started might be treated as a parameter of the (idle) action. Likewise, state changes without an explicit action requirement can occur. For this purpose, we introduce "resume" action as state change on coworker action with persisting own action, which can be any of the action possibilities specified in Section IV-D including "idle" action (transitions to same state or changes state on coworker action)

Principally, we can generate beliefs about action interdependencies from recurring observations in an unsupervised fashion and capture them in the probabilistic model/reward structure, where there is no significant risk associated with breaching the action dependencies (e.g. forget to put the saucer in place before the cup or do not put a placemat on the table before the plate) due to the (at least initial) lack of a high-confidence estimator.

For safety reasons, we want to provide a means of injecting critical constraints between actions using a formal description. In essence, this approach will have to assume and impose deterministic behavior in some parts of the decision making process (shelf must not be released until work is definitely done, or it will fall). Thinking this a bit further (and considering current means of handling safety critical states in industrial applications) it might even be a good choice to define different stages of acknowledgement mechanisms according to the risk assessment (i.e., do not rely on own belief whether work is done, ask for it explicitly and wait for vocal or physical acknowledgement) or (even in uncritical sections) according to the current trust level of the user.

As the action precondition types, we adopt the formalism of Permanent, Enabling and Ordering constraints proposed by [15] (see Figure 2).

- Permanent preconditions must be met before and must be maintained during execution (Pouring task, Securing task)

- Enabling preconditions must be met immediately before the execution but can change during execution
- Ordering constraints must have been met at some point before the action (Switch of power before changing a light bulb)

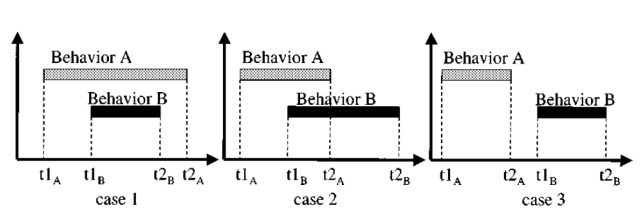

Fig. 2. Temporal precondition types between actions: Permanent (case 1), Enabling (case 2), Ordering (case 3) [15]

The preconditions could e.g. be specified as Temporal Logical Formulas that have to be satisfied. Admittedly, judging these situation will require exhaustive sensing and reasoning capabilities and universal background knowledge regarding safety critical task constraints for not having to hand tailor the entire workflow. We do currently not have an answer to the question how this online risk assessment can be performed or formulated in a generally valid fashion, as risk is often highly dependent on the current environment. However, we think that providing a methodology for respecting those constraints and allow model checking for safety critical sections is especially important in human-robot collaboration tasks.

D. Movement and Action Primitives

A generic action model is required in order to avoid exhaustive training of classifiers and permit mapping to parametric robot skills reflecting the same action. The idea is to use action primitives and a grammatical description of how these primitives can be combined, graphically depicted in Figure 3. According to [16], actions on objects may be described by a general approach-act(use)-remove cycle. Thus, it is possible to identify all human movements and trajectories to be instances of the same primitive as long as they induce the same effect on the object (as specified by the action set as ”MOVE”, ”HOLD”, ”IDLE”, ”USE”/”USE WITH”). To achieve this goal, a suitable feature set and generic classifiers in order to determine the primitives have to be developed.

The input to the ”configuration” step is an RGBD stream from Microsoft Kinect. The camera is set up in some distance (app. 2.5m) to the scenery, reflecting the robot watching humans perform the task. Skeleton streams using the hand track and tracks of all present objects are computed. K-d tree clustering is performed in order to obtain object regions and the COG of each object is computed. Filtering and fusion of object and hand locations is done with an Unscented Kalman Filter at a sample rate of 100 ms. From hand tracks and tracks of all present objects, a couple of simple, computationally inexpensive features (velocity, relative velocity, distance,

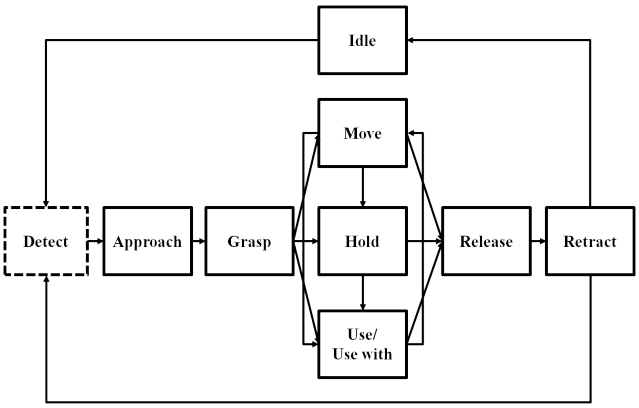

Fig. 3. Graphical depiction of object manipulation transition model

etc.) can be computed for each hand-object and object-object pair. A representation of the overlayed velocity profiles and the respective hand/object distance can be seen in Figure 4. For the human eye, sequences in the interaction streams, like the characteristic bell shaped velocity profiles, are clearly recognizable. On this feature set, a classifier should be trained using common machine learning techniques (e.g. Random Forest). For validation and independent evaluation of the model building methodology, input streams will be manually annotated.

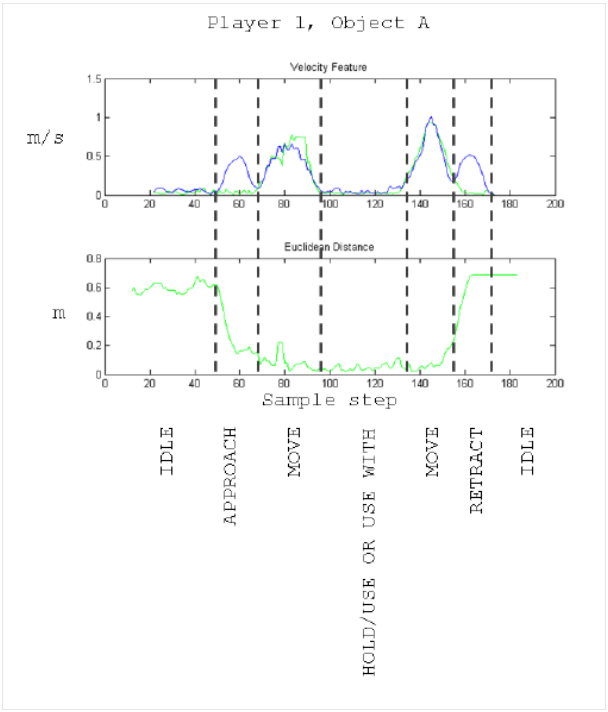

Fig. 4. Hand/object velocity profiles (upper) and hand/object distance evolution (lower)

E. Model Generation

From the object ontology, we can enumerate the complete MOMDP state-space and associate a set of actions and most likely observations (human action, (topological) object

states) according to the restrictions on what can be done with the objects like:

- Tangible objects can be grasped/touched
- All grasped objects can be released
- All movable objects can be moved/held
- All content objects can be "put" into a related object
- All usable objects can be "used" and/or "used with" another object, related effects
- Idle and Resume actions

If, in contrast to our assumption the robot (or human) is not able to perform certain actions, or is especially suited for certain tasks, this might be reflected in the action/observation set. After this step, the potential model space to explore and the possible task orders are very large, if no further restrictions are made. The configuration of the model is based on the observation of interaction sequences and primitive actions as explained in Section IV-D from which we can try to derive the reward function of the model (Inverse Reinforcement Learning).

REFERENCES

[1] C. Breazeal, et al., Effects of nonverbal communication on efficiency and robustness in human-robot teamwork, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05), 2005, pp. 707-713.

[2] G. Hoffman and C. Breazeal, Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team, Proceedings of the ACM/IEEE international conference on Human-robot interaction (HRI '07), 2007, pp. 1-8.

[3] C. Diuk, A. Cohen and M. L. Littman, An object-oriented representation for efficient reinforcement learning, in Proceedings of the 25th international conference on Machine learning, 2008, ACM, pp. 240–247

[4] P. Abbeel and A. Y. Ng, Apprenticeship learning via inverse reinforcement learning. Proceedings of the twenty-first international conference on Machine learning (ICML '04), 2004, pp. 1-.

[5] S. Nikolaidis, et al., Efficient model learning from joint-action demonstrations for human-robot collaborative tasks, Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '15), 2015, pp. 189-196.

[6] S. R. Schmidt-Rohr, M. Loesch and R. Dillmann, Learning flexible, multi-modal human-robot interaction by observing human-human-interaction, RO-MAN, 2010, pp. 582587.

[7] L. P. Kaelbling, M. L. Littman and A. R. and Cassandra, Planning and acting in partially observable stochastic domains, Artificial intelligence 101(1), 1998, pp. 99134.

[8] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, Planning under uncertainty for robotic tasks with mixed observability. IJRR, 2010, pp. 1053-1068.

[9] D. A. Norman. Affordance, conventions, and design. interactions 6, 3, 1999, pp. 38-43.

[10] Y.Zhu, A. Fathi, and F. Li. Reasoning about object affordances in a knowledge base representation. European conference on computer vision, 2014, pp. 408-424.

[11] M.Khamassi, et al., Active exploration and parameterized reinforcement learning applied to a simulated human-robot interaction task, IEEE International Conference on Robotic Computing (IRC), 2017, pp. 28-35.

[12] F. Woergotter, et al., A simple ontology of manipulation actions based on hand-object relations. IEEE Transactions on Autonomous Mental Development, 2013, pp. 117134.

[13] G. J. Maeda, et al., Probabilistic movement primitives for coordination of multiple humanrobot collaborative tasks. Autonomous Robots 41.3, 2017, pp. 593-612.

[14] W. Masson, P. Ranchod, and G. Konidaris, Reinforcement Learning with Parameterized Actions. In AAAI, 2016, pp. 1934-1940.

[15] M. Nicolescu and M. Mataric, Learning and interacting in human-robot domains. Systems, Man and Cybernetics, Part A: Systems and Humans, 2001, pp. 419430.

[16] V. Kruger et.al., Learning actions from observations, IEEE Robotics and Automation Magazine, 2010, pp. 3043.