Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments

Michael Pardowitz, Steffen Knoop, Ruediger Dillmann, *Member, IEEE*, and Raoul D. Zöllner

***Abstract*—Since many years the robotics community is envisioning robot assistants sharing the same environment with humans. It became obvious that they have to interact with humans and should adapt to individual user needs. Especially the high variety of tasks robot assistants will be facing requires a highly adaptive and user-friendly programming interface. One possible solution to this programming problem is the learning-by-demonstration paradigm, where the robot is supposed to observe the execution of a task, acquire task knowledge, and reproduce it. In this paper, a system to record, interpret, and reason over demonstrations of household tasks is presented. The focus is on the model-based representation of manipulation tasks, which serves as a basis for incremental reasoning over the acquired task knowledge. The aim of the reasoning is to condense and interconnect the data, resulting in more general task knowledge. A measure for the assessment of information content of task features is introduced. This measure for the relevance of certain features relies both on general background knowledge as well as task-specific knowledge gathered from the user demonstrations. Beside the autonomous information estimation of features, speech comments during the execution, pointing out the relevance of features are considered as well. The results of the incremental growth of the task knowledge when more task demonstrations become available and their fusion with relevance information gained from speech comments is demonstrated within the task of laying a table.**

***Index Terms*—Human–robot interaction, programming by demonstration, task learning.**

I. Introduction

DURING the last years, humanoid robotics became a major trend in the robotics community. Generally, the service robot design is dominated in all facets by the fact that in the future more and more robots are supposed to act in close cooperation with humans. In order to persist in the depicted scenario, these robots have to autonomously adapt their behavior and actions to the users needs and preferences. Therefore, a very important property of robot assistants is learning, since it enables robots to cope with everyday situations and natural (household or office) environments in a way that pays attention to their users. This means that through the learning process, on the one hand, new skills and tasks have to be acquired, and on the other hand, the knowledge of the system has to be adapted to new contexts and situations.

Robot assistants coping with these demands must be equipped with innate skills and should learn livelong from their users. Supposedly, these are no robot experts and require a system that adapts itself to their individual needs. For meeting these requirements, a new paradigm for teaching robots is defined to solve the problems of skill and task transfer from human (user) to robot, as a special way of knowledge transfer between man and machine.

Obviously, systems capable of this knowledge transfer require the following:

1) powerful sensor systems to gather as much information as possible by observing human behavior or processing explicit instructions like commands or comments;
2) a methodology to transform observed information for a specific task to a robot-independent and flexible knowledge structure;
3) actuator systems using this knowledge structure to generate actions that will accomplish the acquired task in a specific target environment.

This approach to task learning is widely known as programming by demonstration (PbD). A crucial point in task learning is to exactly capture the user's intention. Usually, this is achieved by observing multiple demonstrations of the same task and identifying the common features of the task as the user's inherent intention. On the other hand, initially requesting multiple demonstrations of a single task would annoy the user. Therefore, PbD systems should be capable of learning a task from a single demonstration in order to allow first executions, monitored by the user. Incremental learning approaches that gradually refine task knowledge and generalize it as more demonstrated examples become available pave the way toward suitable PbD systems for humanoid robots.

One aspect that can only be learned incompletely from a single user demonstration is the degree of freedom with respect to the sequence the subparts of a certain task can be scheduled.

Manuscript received December 19, 2005; revised May 11, 2006. This work was supported by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020. This paper was recommended by Guest Editor Y. Demiris.

M. Pardowitz, S. Knoop, and R. Dillmann are with the Institute of Computer Science and Engineering, Universität Karlsruhe (TH), 76138 Karlsruhe, Germany (e-mail: pardow@ira.uka.de; knoop@ira.uka.de; dillmann).

R. D. Zöllner is with Siemens Corporate Technology, Information and Communications, 81739 Munich, Germany (e-mail: raoul.zoellner@siemens.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2006.886951

This knowledge allows the robot system to execute them in a sequence that can be optimized with respect to speed, energy consumption or path length. In order to learn these aspects from multiple demonstrations, common subtasks of different demonstrations of the same task have to be identified. This needs an explicit measurement of subtask similarity, that in turn relies on the similarity of features of each subtask. Different features have different importance attached. Here, an approach is proposed that allows to estimate the relevance of a feature, based on background knowledge, its occurrence probability in all demonstrations of a certain task and vocal comments the user can give while demonstrating the task to be learned by the system.

The remainder of this paper is organized as follows. The next section gives an overview on related work concerning PbD and task learning from user demonstrations. Section III describes the system for the acquisition of task knowledge from a single user demonstration and the general representation of task knowledge, called macrooperators (MOs). Section IV introduces task precedence graphs (TPGs), the representation for the sequential structure of a task and proposes a method for reasoning on the underlying precedence graph of a task with two or more sequential demonstrations given. The methods for identifying corresponding subtasks within different task demonstrations described in Section V are an important preliminary for this computation. Heuristical relevance estimates for task features based on the current task demonstrations, past experiences, and spoken user comments are used to compute the similarity of tasks and subtasks and will be discussed in Section VI section. Finally, the methods described in earlier sections are evaluated in Section VII.

II. Teaching Robot Assistants: An Overview

A classification of methods [1], [2] for teaching robot assistants in an intuitive way might be done according to the type and granularity of knowledge which is supposed to be transferred between human and robot. Here, the proposed methods for learning from human activities can be categorized into subsymbolic skill learning systems (see Section II-A) and symbolic task acquisition systems (see Section II-B). Work specially focusing the analysis of sequences of actions is treated in Section II-C.

A. Skill Learning

For the term "skill," different definitions exist in literature. In this paper, it will be used as follows. A skill denotes an action (i.e., manipulation or navigation), which contains a close sensor-actuator coupling. In terms of representation, it denotes the smallest symbolic entity which is used for describing an action.

Examples for skills are grasps or moves of manipulated objects with constraints like "move along a table surface," etc. The classical "peg in hole" problem can be modeled as a skill, for example using a force controller [3]. Skill learning means to find the control function π, which satisfy the equation $U(t) = \pi(X(t),\ Z(t))$ and transfers sensor data and internal states to actuator output. The model of the skill transfer process is visualized in Fig. 1 [4] reports successful learning of advanced skills through reinforcement learning. The robot system is given the demonstration of a pendulum swingup, a human challenging skill, and is able to reproduce that within 4–5 trials.

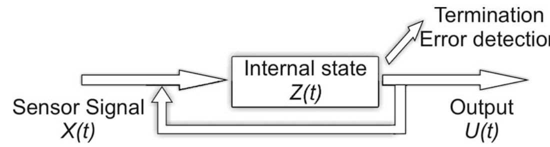


Fig. 1. Process of skill transfer.

Successful learning of gestures is reported in [5]. The aim is to reproduce trajectories of movements like waving or drawing letters, aiming at a high similarity of the robot's movements to the demonstrated trajectories, applying different imitation metrics [6].

However, teaching skills to robot assistants is time intensive and affords a lot of knowledge about the system. Therefore, looking at robot assistants were the user of the systems is not an expert, it seems more feasible that the majority of the needed skills are prelearned. The system would have a set of innate skills and will only have to learn a few new skills during its "life time."

B. Task Learning

From the semantic point of view, a task denotes a more complex action type than a skill. From the representation and model side a task is usually seen as a sequence of skills. In terms of knowledge representation, a task denotes a more abstract view of actions, where the control mechanisms for actuators and sensors are encapsulated in modules represented by symbols.

Various task learning systems have been proposed with various sensor and actuator settings on different task domains like assembly work [7], block world construction [8], [9] or household manipulation [10]. In spite of all differences, a similar task model is evincing.

A task T is understood as a sequence of skills [or elementary operators (EOs)] S_i, the context including environmental information or constraints E_i and internal states of the robot system Z_i are introduced as pre- and postconditions of the skills. Formally, a task T can be described as

$$\begin{aligned} T &= \{(\text{PreCond}(E_i, Z_i))\, S_i\, (E(E_i, Z_i))\} \\ E &= \text{PostCond} \setminus \text{PreCond} \end{aligned} \tag{1}$$

where E denotes the effect of a skill on the environment and the internal robot states. Consequently, the goals and subgoals of a task are represented by the effect of the task or the subsequence of the skills.

A noticeable difference of tasks from skills is the fact that tasks are state, but not time dependent like skills, and therefore enable a higher generalization in terms of learning. Although the sensory-motor skills in this representation are encapsulated in symbols, learning and planning task execution for robot assistants becomes very complex due to the fact that state

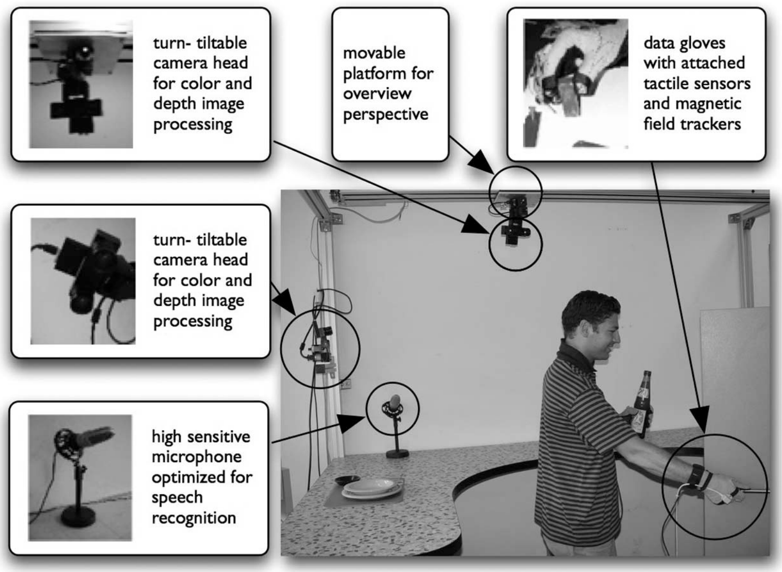


Fig. 2. Training center with dedicated sensor environment.

descriptions of environments (including humans) and internal states highly complex.

C. Learning of Sequences of Actions

Sequential analysis has been applied to both the skill and the task level.

1) Chen and Zelinsky [11] records multiple user demonstrations of a complex manipulation skill of inserting a spindle into supports. A minimal sequence of motions is learned from five demonstrations, minimal in the sense of the necessary succession of actions. Unnecessary actions that do not appear in all demonstrations are pruned and only the sequence of essential actions is retained.
2) Nicolescu and Mataric [12] stresses the role of verbal interaction and feedback with and from the human teacher to facilitate learning of sequential arrangements of behaviors. A navigation task for a Pioneer 2-DX mobile robot was chosen as the task domain. The resulting task description after several demonstrations contains mainly the longest subsequence of navigation behaviors common to all demonstrations of the task.
3) Sutton *et al.* [13] covers temporal abstraction in reinforcement learning of a two-layered task model. The task domain is also from the navigation domain, dealing with movement behaviors in multiroom indoor environments. Several demonstrations of reaching a certain point in a certain room from different starting positions are generalized to a behavior leading to the destination from anywhere in the building.

Concluding from the methods presented in this section, one can say that there is no general representation featuring both sequential and concurrent activities on the task level, as well as no methods for incremental and multimodal task learning in the household robot domain. This need is met by the sequential task representation proposed in Section IV and the interactive task feature learning mechanisms in Section VI.

III. System Architecture and Task Model

The starting point for acquiring new task knowledge is a PbD system developed for many years at our lab. This section gives a brief overview of the segmentation process and the manipulation task classes which can be learned by the system in order to describe the gathered information and the assumptions made during this process.

The main idea is to separate the acquisition of task knowledge from the execution of the task on a specific robot in order to achieve a robust task description from a vast external sensor system installed in a so-called training center. The training center is an environment for capturing human activity. For experiments, a kitchenlike environment was set up where the user may perform typical tasks such as loading a dishwasher, depositing objects in a refrigerator, opening bottles, filling glasses with liquid, etc. Since robustness and flexibility are important for its acceptance, the workspace resembles that of a real kitchen and various sensors are integrated for stable observation (see Fig. 2). Table I contains a list of these sensors and their output.

For ensuring a representation of robot invariant task knowledge the observed demonstration will be transformed to an abstract hardware-independent stripslike tree description called MO relaying on basic skills (see Section II-A) called EOs. An EO denotes an action that abstracts a sensory motor coupling like a grasp or a linear move, etc. The EOs build the hardware abstraction layer and have to be implemented on the robot system depending on the available sensors and actors.

A. Classes of Manipulation Tasks

Within the framework of PbD, the basis for successfully mapping task solutions from a human to a robot system is the ability of the programming system to recognize and to interpret human actions. It is important to note that humans tend to perceive activity as a clearly separated sequence of subtasks,

TABLE I
SENSORS AND OBSERVABLE FEATURES IN TRAINING CENTER ENVIRONMENT

sensor	observable feature
turn- and tiltable camera at head level	front view of the human, hand gestures, focus of attention, manipulatable objects
turn- and tiltable camera at ceiling level, mounted on a linear drive	upper body, limb gestures, torso motion, refridgerator contents (shoulder view)
data gloves	static and dynamic grasps with according grasp types
magnetic field sensors	hand trajectories
highly sensitive microphone	speech commands and comments

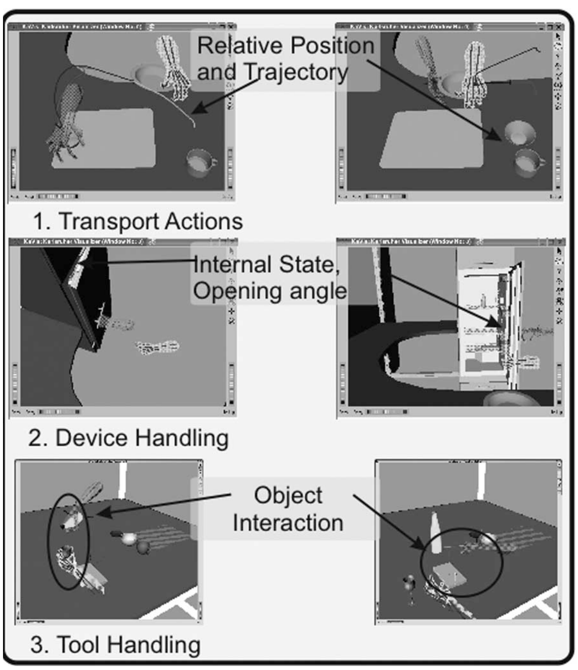


Fig. 3. Examples for the manipulation classes distinguished in a PbD system.

each of which directly realizes a specific subgoal [14]. The solution of the overall problem is representable as a sequence of such subtasks.

The most useful information for interpreting such a sequence is to be found in the subgoals of each manipulation segment, i.e., their contribution to the effect of the demonstration. Three different classes of functional roles with different intended subgoals can be distinguished (see Fig. 3).

1) *Transport operations*: are the simplest action classes of robots in the role of servants or assistants. The change of the external state (Cartesian position) of the manipulated object serves as a formal distinguishing criterion of this kind of task. Tasks like pick and place or fetch and carry denoting the transport of objects are part of almost all manipulations. Consequently, for modeling transport actions, the trajectory of the manipulated object has to be considered and modeled. In terms of teaching transport actions to robots, the acquisition and interpretation of the performed trajectory is therefore crucial.
2) *Device handling*: A more specialized class of manipulation tasks deals with changing the internal state of objects without the influence of other objects (like opening a drawer, pushing a button, etc.). Every task changing an internal state of an object without manipulating another object belongs to this class. Actions of this class are typically applied when using devices, but many other objects also have internal states that can be changed (e.g., a bottle can be open or closed, filled or empty, etc.). In terms of modeling device-handling tasks, transition actions leading to an internal state change have to be modeled. Additionally, the object models need to integrate an adequate internal state description. Teaching this kind of task requires observation routines able to detect internal state changes by continuously tracking relevant parameters.
3) *Tool handling*: The most distinctive feature for actions belonging to the class of tool handling is the interaction between two objects, typically a tool and some workpiece. The interaction is related to the functional role of objects used or the correlation between the functional roles of all objects included in the manipulation, respectively. The object model thus should contain a model of the possible interaction modalities or functional roles the object can take. According to different modalities of interaction, considering contact, movements, etc., a diversity of handling methods has to be modeled. In terms of teaching robots, the observation and interpretation of the actions can be done using parameters corresponding to the functional role and the movements of the involved objects.

These three distinguished classes cover the domain of performative actions within household and workshop environments, and allow to model and interpret human actions with respect to the inherent semantics pursued. The following section deals with the issue of how these three classes can be automatically segmented from human demonstrations.

B. Segmentation of the Demonstration

The phases of the PbD process, described in [15], are a segmentation of the sensor data followed by an analysis and interpretation step for identifying a sequence of EOs which is abstracted and generalized to a MO in a further step.

In order to process the above manipulation classes, the segmentation of the sensor data is done in three phases.

1) *Grasp segmentation:* Here, a stable segmentation of grasp and release actions is performed.
2) *Trajectory segmentation:* This phase extracts for each grasp and release segment an approach and depart trajectory and fragments the whole hand trajectories in elementary move operations like linear, or circular segments.

TABLE II
PARAMETERS FOR BASIC SKILL SEGMENTATION

Basic Skill	Parameter
Transport:	
Grasp:	Hand:
static	TCP velocity, joint angles
dynamic	+ forces
Move types	TCP trajectory analysis
Device handling:	+ Object model (Type)
Open doors, drawers	Move-axis, handle, state
Push/rotate buttons	Move-axis, handle, state
Tool handling:	+ Functional Role
Screwing	”’Screw-able”’
Pouring	”‘pour in /out”’

3) *Statistic segmentation:* For detecting object relations during grasp phases, statistic parameters are used in order to generate hypothesis on possible interactions using object-role-dependent probabilities [16].

The segmentation of the user demonstration uses a lot of background knowledge about the manipulation process and the environment, respectively, the objects used in the demonstrated manipulation. In Table II, the parameters used for the segmentation are listed. The table shows the dependence of the parameters from the manipulation class. For simple transport actions, in addition to the grasp type, only the trajectory of the objects is needed. Looking at device handling tasks, where the manipulation changes the internal state of the manipulated object (i.e., for an "open a door" action the state "door closed" changes to "door open") more information about the object has to be included in the process. The most complex manipulation in terms of detection is a tool handling since in this case the interaction between the manipulated objects determines the goal of the action. In this case, more information about the object types and their functional roles is needed in order to detect and describe this kind of manipulations.

The result of the segmentation step is a list of key points which are indicating possible starting points of EOs. For generating an action (EO) sequence for each fragment a search for an instantiation of EOs is made. The search is triggered by the information about the type of these key points and a confidence measure for this classification. The proof for the instantiation of an EO requires an evaluation of several EO conditions stored in the EOs. For example, for instantiating a static grasp, a neuronal net is used in order to classify the grasp and a positive classification will result in an instantiation of a certain static grasp, i.e., circular grasp (see [17]).

C. Hierarchical Task Representation

Representing manipulation tasks as pure action sequences is not flexible and also not scalable, especially when they are used as input for planning systems (see [18]). Therefore, a hierarchical representation is introduced in order to generalize an action sequence according to the explanation-based theory

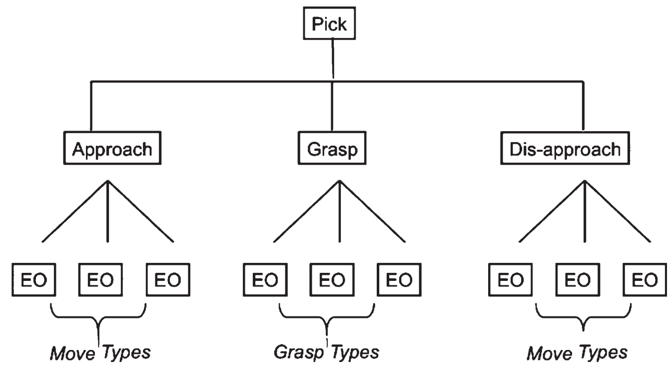


Fig. 4. Hierarchical model of a pick action.

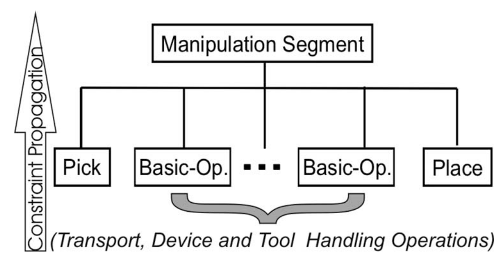


Fig. 5. Representation of a manipulation segment.

[19] from a single demonstration and to combine EOs to more complex compound tasks.

Looking only on manipulation tasks the assumption is made that each manipulation consists of a grasp and a release action. To cope with the above specified manipulation classes, pushing or touching an object is interpreted as a grasp. The representation of grasping an object constitutes a "pick" action and consist of three subactions: an approach, a grasp type, and a disapproach (Fig. 4). Each of these subactions consists of a variable sequence of EOs of certain types (i.e., for the approach and disapproach the EOs are move types and the grasp will be of type grasp). A "place" action is treated analogously.

Between a pick and a place operation, depending on the manipulation class, several basic manipulation operations can be placed (Fig. 5). For example, a demonstration of the task "pouring a glass of water" consists of the basic operations: "pick a bottle," "transport the bottle," "pour in," "transport the bottle," and "place the bottle." A sequence of basic manipulation operations starting with a pick and ending with a place is abstracted to a manipulation segment.

The level of manipulation segments denotes a new abstraction level on closed subtasks of manipulation. In this context, closed means that segments start and end with both hands free and that the environmental state is stable. Furthermore, the synchronization of EOs for left and right hands are included in the manipulation segments. Pre- and postconditions describing the state of the environment at the beginning and at the end of a manipulation segment are sufficient for their instantiation. The conditions are propagated from the EO level to the manipulation segmentation level and are computed from the environmental changes during the manipulation. In parallel to the propagation of the conditions, a generalization in terms of positions and object types and features is done.

In this section, a system, capable of one-shot learning of complex household task has been presented. The resulting hierarchical task description called MO contains subtasks and pre- and postconditions and is suitable as input for symbolical task planning systems. So far, the system only takes into account a single task demonstration and fails to generalize its task knowledge when multiple demonstrations become available. How this can be done is presented in the following section.

IV. Learning of TPGs

This section is concerned with the representation of the sequential features of a task and how it can be learned from multiple user demonstrations of the same task. A formal definition of the structure that contains the sequential ordering a task obeys is given. Additionally, a way a system can make hypotheses about the sequential dependencies is presented.

When a user performs a task, he performs its subparts (the manipulation segments) in a sequential ordering that he has chosen by random or by intent from all possible task execution orders. For a system recording his actions, these appear as a simple sequence of operations. In order to exploit the maximum degrees of freedom, the task possesses at execution time, the sequential constraints a valid task execution has to obey must be known to the robot before the execution can start. Observing a single demonstration of a certain task, the sequence chosen by the user is influenced by two different aspects.

1) Sequential dependencies induced by the task to be solved. These form temporal precedence relations that follow from the attributes of the task to be done and environmental constraints. One simple example is the task of fetching an object from the fridge. The subtask of opening the fridge door must be accomplished before a robot can pick the object.
2) Sequential execution of independent operations. Operations sequentially independent of each other that cannot be performed in parallel have to be executed in any order. This order may be chosen by user's preferences or according to any strategy or optimization criteria.

A learning system that builds knowledge about a task has to make hypotheses about the underlying sequential task structure. These hypotheses can be represented by TPGs.

Definition 1: A TPG for a task T is a directed graph $P = (\mathbf{N}, \mathbf{R})$ with $\mathbf{N}$ being the set of subtasks $o_1, o_2, \ldots, o_n$, and $\mathbf{R} \subset \mathbf{N} \times \mathbf{N}$ being the set of precedence relations a faultless task execution must comply with. A precedence relation $(o_1, o_2) \in \mathbf{R}$ with $o_1, o_2 \in \mathbf{N}$ implies that the operation o_1 must be complete before the execution of o_2 can start. This is abbreviated as $o_1 \rightarrow o_2$.

A faultless execution of a task requires the chosen sequence of operations to be consistent with its TPG, or, in other words, fulfills every sequential relationship inherent to the TPG.

Definition 2: A demonstration $D = (o_{i_1}, o_{i_2}, \ldots, o_{i_n})$ with $o_{i_j} \in \mathbf{N}$ is said to comply with a TPG $P = (\mathbf{N}, \mathbf{R})$, if for every precedence relation $o_i \rightarrow o_j \in \mathbf{R}$ the operations $o_i = o_{i_k}$ and $o_j = o_{i_l}$ appear in the correct sequential order, i.e., $k < l$. This is denoted by $P \rightsquigarrow D$.

For a system that is supplied with only a single user demonstration $D = (o_1, o_2, \ldots, o_n)$ of a task, it is hard to guess the inherent TPG. Suppose that an operation o_i is observed before o_j. Then, $o_i \rightarrow o_j$ can be part of the TPG, indicating that in every valid task execution sequence o_i must appear before o_j, or this observation can result from the fact that the user had to chose any ordering for two sequential independent actions (see above). The learning system can state different hypotheses, ranging from the most restrictive TPG $P^D = (\mathbf{N}, \mathbf{R}^D)$ with

$$\mathbf{R}^D = \{(o_i, o_j) | i < j\} \tag{2}$$

to the TPG with the most degrees of freedom, $P^* = (\mathbf{N}, \emptyset)$. P^D restricts the task to be executable only with the sequential ordering observed in the user demonstration, P^* classifies every order of actions as a valid task sequence. All other possible TPGs the user demonstration complies with are valid as well. In order to impose a structure on this set of valid hypotheses, the "more general" partial ordering is defined.

Definition 3: A TPG $G = (\mathbf{N}, \mathbf{R}_G)$ is said to be more general than a TPG of the same task $S = (\mathbf{N}, \mathbf{R}_S)$ if and only if $\mathbf{R}_G \subset \mathbf{R}_S$. This is abbreviated as $G \succ S$.

The remainder of this section deals with the topic of how a valid hypothesis that complies with all task demonstrations can be learned in an incremental way by using the more general relation.

While the learning system cannot decide which TPG from the set of consistent TPGs fits the task best after seeing only one single example, it seems a viable approach to supply it with more sample demonstrations, applying different task execution orders. In order to learn task knowledge from even a single demonstration sufficient for execution but improving the learned task when more knowledge in form of task demonstrations is available, an incremental approach is chosen.

Assuming that the system has learned the most specific TPG P_m after obtaining a set of m Demonstrations $\{D_1, D_2, \ldots, D_m\}$, a new demonstration of the same task D_{m+1} is observed. The next step is to adapt the learned TPG in a way that incorporates the new knowledge. Mitchell [20] suggests that this can be done by further generalizing the previous hypothesis to a new one, covering the additional example. In order not to generalize too far, that is, to ensure that no essential precedence relations are dropped, the minimal generalization of P_m that is consistent with D_{m+1} must be chosen. Therefore, one can state that the best choice for P_{m+1} is the hypothesis H with

$$H \succ P_m \wedge H \rightsquigarrow D_{m+1} \wedge (\nexists H' : H' \rightsquigarrow D_{m+1} \wedge H \succ H' \succ P_m). \tag{3}$$

In computation terms, the new set of task precedence relations $\mathbf{R}_{m+1}$ can be expressed as a function of the previously learned hypothesis $P_m = (\mathbf{N}, \mathbf{R}_m)$ and the most restrictive hypothesis for the new observed demonstration $P^{D_{m+1}} = (\mathbf{N}, \mathbf{R}^{D_{m+1}})$, which can be computed according to (2)

$$\mathbf{R}_{m+1} = \mathbf{R}_m \cap \mathbf{R}^{D_{m+1}}. \tag{4}$$

Now, one can state the process of incremental learning of TPGs.

1) For the first training example D_1, initialize the TPG $P_1 = P^{D_1}$ according to (2).
2) For each additional demonstration D_{m+1} of the same task, compute $P^{D_{m+1}}$ and update the hypothesis P_{m+1} according to (4).

One issue not addressed until now is the question of when the additional task demonstrations arrive. In the application domain of household service robots, one cannot assume that the user gives all demonstrations sufficient to learn the correct TPG at once. Instead, it might take a long time between two successive demonstrations of the same task. Moreover, the task knowledge learned during past demonstrations has to be utilized because it is likely that the user will require the robot to execute the learned task without having taken into account all task demonstrations that might appear in the future. Therefore, the TPG learned by the system should be reliable enough to ensure a faultless and, above all, secure task execution.

The incremental learning mechanism for TPGs presented in this section allows a correct precedence graph to be learned from even a single example while still maintaining the ability to incorporate new knowledge in order to refine the task knowledge and to provide additional reordering opportunities at execution time.

The user is given the possibility to provide the learning system with another task demonstration, when during a robot's task execution he finds out, that the learned TPG is to restricted to meet his intention.

V. Recognition of Common Subtasks Applying Subtask Similarity Measures

While the last section presented a method for learning the sequential constraints of a task when in every task demonstration exactly the same subtasks are used, this is not likely to be true for every task. Usually, the human demonstrator will only use similar but partly different manipulation segments in order to fulfill the same task. This section deals with the topic of identifying the corresponding manipulation segments in two or more demonstrations of the same task.

In order to recognize the matching manipulation segments for two different task demonstrations, the ordering of the manipulations cannot be a reliable measure for subtask correspondence as the sequence of subtasks performed is potentially permuted. Although matching the segments that manipulate the same class of objects seems to be a good idea at first, this method fails as soon as there are multiple objects of the same class present in the scene. Therefore, more features should be taken into account to determine the similarity of subtasks and establish sufficiently robust subtask correspondences in order to identify operations of equal impact to the scene.

The features of a manipulation segment are organized along the following classification.

1) *Object features*: These contain the class of the objects manipulated or referenced in the certain subtask (cup, plate, table, etc.) as well as their possible functional roles (liquid container, object container, etc.).
2) *Pre- and postcondition features*: These contain the geometrical relations that exist between the objects before or after the performance of the subtask, respectively.

The base operation to assess feature conformance is the measure of similarity s_F for two sets of features $\mathbf{A}, \mathbf{B}$ with

$$s_F(\mathbf{A}, \mathbf{B}) = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|}. \tag{5}$$

This is the portion of common features in both feature sets.

It turned out that this straightforward approach exposes one major drawback: As the number of features is relatively high (about 250–300 features for tasks dealing with 3–5 objects) and many of the features are irrelevant to the certain task to be learned, the features carrying the relevant information will be predominated by the irrelevant ones. The solution is to weight each feature f depending on it is relevance to the task to be learned with a weight $w(f)$. This turns (5) into

$$s_F(\mathbf{A}, \mathbf{B}) = \frac{\sum_{f \in \mathbf{A} \cap \mathbf{B}} r(f)}{\sum_{f \in \mathbf{A} \cup \mathbf{B}} r(f)}. \tag{6}$$

The weight function $r(f)$ could be influenced by several different aspects like background knowledge, information content of features, and vocal elucidations; the user gives during the demonstration of a task. This issue is discussed in greater detail in the following section. For now, it is assumed, that a weight function $r(f)$ exists that somehow reflects the importance of a feature.

With the weighted proportion of features s_F as in (6) it is possible to compute the similarities s_{pre} and s_{post} of the pre- and postcondition sets of two subtasks M_1 and M_2 and the average of the subtask similarities $\overline{s_{\text{sub}}}$ of the tasks. They are set to

$$s_{\text{pre}} = s_F(\text{Precond}(M_1), \text{Precond}(M_2))$$

$$s_{\text{post}} = s_F(\text{Postcond}(M_1), \text{Postcond}(M_2))$$

$$\overline{s_{\text{sub}}} = \frac{1}{|M_1 \cup M_2|} \sum_{m_1 \in M_1, m_2 \in M_2} s_M(m_1, m_2).$$

The (recursive) overall similarity s_M of the two manipulation segments M_1, M_2 is defined as the weighted average of the pre- and postcondition similarity and the average of all subtask similarities

$$s_M(M_1, M_2) = \alpha_{\text{pre}} s_{\text{pre}} + \alpha_{\text{post}} s_{\text{post}} + \alpha_{\text{sub}} \overline{s_{\text{sub}}}.$$

In the experiments conducted in this paper, these weights were normalized to 1/3. Once the subtask-similarity s_M is computed for every pairing of the task's subtasks, the subtask correspondence p_{ST} is computed as the permutation that maximizes the sum of similarities

$$p_{\mathrm{ST}} = \arg \max_{p \in \mathrm{perm(subtasks)}} \sum_{(M_1, M_2) \in p} s_M(M_1, M_2).$$

With the subtask permutation p_{ST} the most restrictive hypothesis P^{D_i} (see Section IV) on the underlying task precedence structure can easily be constructed. This hypothesis can then be utilized to learn sequential task constraints in the way described in Section IV.

VI. Incremental Estimation of Feature Relevance Using Vocal Comments and Past Experiences

This section is concerned with the estimation of the relevance of every feature to the task to be learned. Several information channels can be used to guide this weighting process. This section takes into account the physical demonstration and the vocal communication channel. The first provides the representation of actions as described in Section III. Here, task specific knowledge (the feature occurrence probabilities over all demonstrations of the same task) as well as background knowledge (the feature occurrence probabilities over all demonstration the robot has seen in his "lifetime") can be applied.

One way to assess the relevance of a feature is to test its occurrence over several different demonstrations of the same task. Features with a high relevance to the task to be performed have a great probability of occurrence, while features of lower relevance will occur less frequently. According to Shannon, the information content $I(f)$ of a feature f with probability $p(f)$ is $I(f) = -\log_2 p(f)$. As features with low information content (= high probability of occurrence) to the specific task class $\mathbf{T}$ should be favored, $-\log_2(1 - p(f|\mathbf{T}))$ seems a reasonable choice for the weight function.

On the other hand, one has to take into account that when a completely new task is learned or only few demonstrations of the same task are known so far, only little or no information about the distribution of features in the specific task class is known in advance, so the weight function from the last paragraph will not produce reasonable results. The idea is to introduce global background knowledge on the feature distribution. When several demonstrations of different tasks have been observed by the robot and incorporated into the task knowledge base, it can be assumed that the features that uniquely discriminate the task are those that have low occurrence rates across the whole task knowledge base. Thus, the information content to all other tasks $\overline{\mathbf{T}}$ of feature f is $-\log_2 p(f|\overline{\mathbf{T}})$.

An incremental learning system should be able to cope with both situations: with no or sparse knowledge about the specific task features at the initial learning steps as well as with more task demonstrations coming available in later stages. Therefore, a combination has to be found that favors the global information content measure when no or few task demonstrations are available, and the task-specific relevance measure as more demonstrations of the specific task become available. The (partial) weight function for the assessment of feature relevence-based solely on the current task and past experiences that fulfills those requirements is

$$w_{\mathrm{demo}}(f) = -\left[\alpha \log_2\left(1 - p(f|\mathbf{T})\right) + (1-\alpha)\log_2 p(f|\overline{\mathbf{T}})\right] \tag{7}$$

with the relative weighting α of the task-specific knowledge as

$$\alpha = 1 - e^{-k\cdot|\mathbf{T}|}.$$

In our experiments, we chose $k = -(\ln 0.25)/(5)$, such that after observing five demonstrations of the same task, the proportion of task specific to prior knowledge is $0.75:0.25$ and converging toward 1 for more demonstrations. This choice is motivated by [21], stating that the important features of a task can be sufficiently learned from feature occurrence statistics with five demonstrations. This weight function should be normalized to one, resulting in the relative importance of each feature to the task

$$r_{\mathrm{demo}}(f) = \frac{w_{\mathrm{demo}}(f)}{\sum_{f' \in \mathbf{F}} w_{\mathrm{demo}}(f')}. \tag{8}$$

When, additionally, vocal comments are available that correspond with certain features, the estimate of (7) can be further improved. In our system, comments have the form of "telling the system what is going on." For example, when laying the table, the user can tell the system that he/she is putting the fork to the right of the plate on the table. This enables the system to guess that the effect of putting the fork to the right side of the plate is of higher interest to the task than several other features.

Assuming that the evaluation of speech comments resulted in a set $\mathbf{V}$ of features the user reputes important, one can state the vocal part $w_{\mathrm{voc}}(f)$ of the weight function as

$$w_{\mathrm{voc}}(f) = \begin{cases} 1, & f \in \mathbf{V} \\ 0, & f \notin \mathbf{V}. \end{cases}$$

This focuses the system on the features the user highlighted during the demonstrations by giving the vocal comments, while ignoring the others. Again, this is transformed into the relative importance function

$$r_{\mathrm{voc}}(f) = \frac{w_{\mathrm{voc}}(f)}{\sum_{f' \in \mathbf{F}} w_{\mathrm{voc}}(f')}.$$

Combining these two functions r_{demo} and r_{voc} is a crucial issue. It is unclear *a priori*, whether the system should focus on the vocal comments or the actual demonstrations. The first are potentially incomplete as the user comments only parts of the task while he/she rates others as clear and not in need of a comment. On the other hand, the estimate of w_{demo} may be too conservative and unreliable, especially during the first demonstrations of the task, when only little information on the feature distribution of the task class is available.

We propose an approach to this relying on the information content [22]. As we are interested in the features that have high relative weights rather than the ones that are estimated to have low relevance to the task, $-\log(1 - r(f))$ seems to be a good choice (see above). The information contents I_{voc} and

Fig. 6. Learned task of laying the table: final object configuration to be learned.

I_{demo}, respectively, for each input modality are calculated as follows:

$$I_{\mathrm{voc}} = -\sum_{f \in \mathbf{F}} \log\left(1 - r_{\mathrm{voc}}(f)\right)$$
$$I_{\mathrm{demo}} = -\sum_{f \in \mathbf{F}} \log\left(1 - r_{\mathrm{demo}}(f)\right).$$

Now, the final step is to design the overall weight function $r(f)$ as follows:

$$r(f) = I_{\mathrm{voc}} \cdot w_{\mathrm{voc}}(f) + I_{\mathrm{demo}} \cdot w_{\mathrm{demo}}(f). \tag{9}$$

The result in (9) presents an assessment of task features' relevance depending on the their occurrence in all demonstrations of the task, vocal comments, and explanations given during the tasks demonstrations and background knowledge in form of demonstrations of other tasks already provided to the system. All these different information channels are fused taking into account their information content (the amount by which the system can benefit from that information), providing sound foundations for the measurement of task and subtask similarity and higher level learning, e.g., of TPGs.

VII. Results

This section reports and evaluates the experiments with the system described in the preceding sections and analyses its results.

The experiment consisted of teaching an everyday household task from the domain of laying the table. The task for the system to learn was to arrange the objects in a way that can be seen in Fig. 6: A table should be laid on the silver place mat with a plate, a fork to the right of it, and a cup behind the fork (from the users point of view).

Four different demonstrations of the task were provided to the system. In the first two demonstrations, the plate, the fork, and the cup were placed in this order. In the third demonstration, the first two operations were reversed, while in the last demonstration the cup was placed first, followed by the fork and the plate. The following vocal comments were accompanying the demonstrations: In the first two demonstrations, the user told the system "I am putting the plate on the silver place mat, the fork on the silver place mat to the right of the plate and the cup on the mat to the west of the fork." During the third and forth demonstration the user said that he is putting the fork on the mat, the plate to the left of the fork and the cup on the mat in a way, that the fork is to the east of the cup.

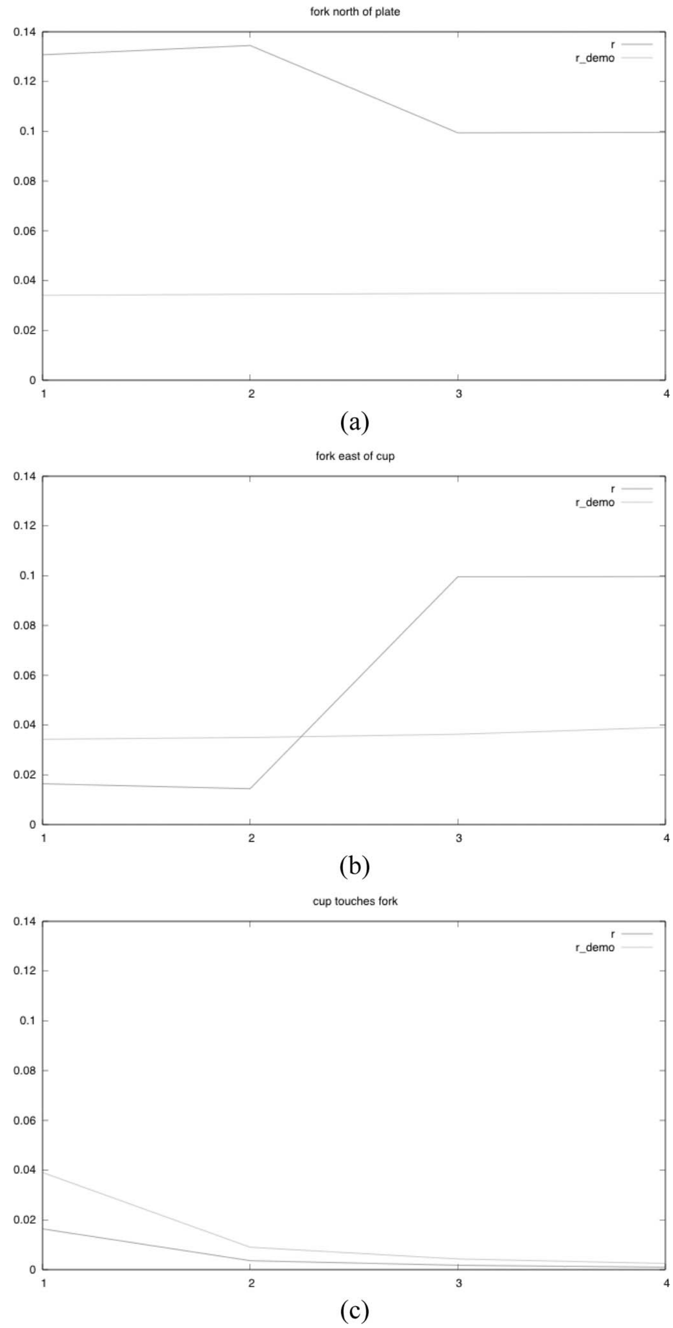


Fig. 7. Weight functions $w(f)$ and $w_{\mathrm{demo}}(f)$ for certain features over number of task demonstrations.

The system could extract seven effects of the demonstrations that were highlighted by the vocal comments. These are "plate on place mat," "fork on place mat," "cup on place mat," "fork north of plate," and "cup west of fork" after the first two demonstrations and additionally "plate south of fork" and "fork east of cup" after the third demonstration.

In Fig. 7, the relative weight function r_{demo} [see (8)], taking into account only the demonstrations, and the compound

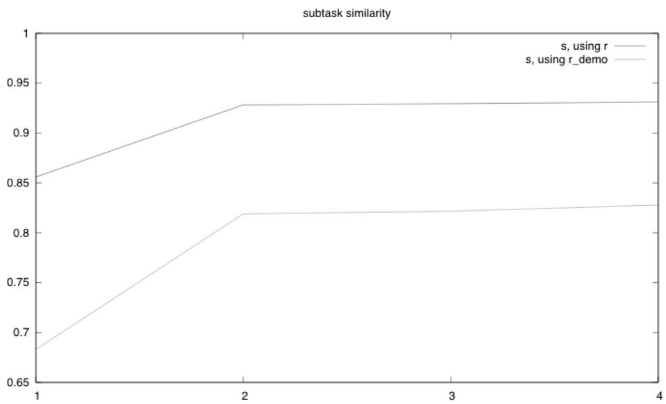
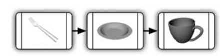
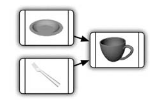
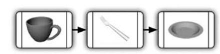


Fig. 8. Subtask similarity measure for the subtask of putting the plate on the place mat over the number of demonstrations.

TABLE III
TPGs Learned Incrementally During the Experiment. The P^{D_i} Column Lists the Most Restrictive TPGs Learned From Single Demonstrations [See (2)], While the P_i Column Shows the Generalized TPGs Learned From All Task Demonstrations the System Has Seen so Far (See Section IV for Details)

i	P^{D_i}	P_i
1 & 2		
3		
4		

relative weight function r that combines r_{demo} with the information from the vocal information channel [see (9)] are plotted over the number of task demonstrations given for different relations. Fig. 7(a) shows the weight function for a very relevant effect: The fork should be put to the north of the plate. One can see that the combined weight function r yields much larger results for the relevance estimate than the weight function r_{demo} deduced from the demonstrations alone. The interesting decrease of the relative weight after the third demonstration results from the two additional relations that occurred in the user comments, distributing the overall weight of 1 over seven instead of five effects in the vocal comment weight function r_{voc}. Fig. 7(b) shows the weight functions for the relation "fork east of cup." In the first two demonstrations, this effect was not mentioned by the user, so the weight is pretty low. As soon as he highlights it in the third demonstration, the relation's weight rises. This shows the advantage of incremental learning mechanisms that can adapt their hypotheses stated so far by using additional knowledge in form of demonstrations or comments. Fig. 7(c) shows the decrease of weight for an irrelevant effect, that was only present in the first demonstration by mistake. As it is not stressed by the user comments and does not appear in later demonstrations, its weight decreases with every user demonstration that is given to the system.

The improvement of the weight function applying vocal comments to the subtask similarity measures [see (6)] is shown in Fig. 8. One can see clearly that the subtask similarity is far greater, leading to more reliable subtask correspondences, which are essential for reidentifying common subtasks during the learning of TPGs.

The first two demonstrations given to the system do not result in a TPG different from the order of subtasks, as no alternative sequences can be observed (see the first row of Table III). After the third demonstration has been observed, the system can apply the reasoning mechanisms on sequential independencies presented in Section IV and obtains the knowledge that the subtasks of putting the plate on the place mat and putting the fork to the right of it are sequentially independent of each other (see second row). After the last demonstration of the task is presented, the TPG is completely learned. The system now obtained the knowledge that each of the three subtasks can be performed independently of each other, that is during execution the three subtasks can be scheduled in parallel or in any order.

VIII. Conclusion

Based on hierarchical, functional, and goal-oriented task models it is possible to acquire and structure task knowledge from a human demonstration. The presented task models and representations are restricted to manipulation tasks, which contain at least one grasp operation. According to the explanation-based theory task knowledge is acquired from a single demonstration. The second part of this paper tackles the problem of structuring the acquired task knowledge trough reasoning. Hereby the concept of TPGs is introduced in order to generalize task knowledge according to the order of subtasks. The crucial point hereby is finding similarity measures for subtasks or actions and since these are based on features the relevance of the features for a certain task have to be estimated. This is done by using the fusion of two information channels namely estimating the relevance of features similar to the rule induction using the information theory and deriving relevant information from speech comments. Finally, a brief discussion of the results shows how the task knowledge over lying a table is acquired and processed.

Acknowledgment

This work has been partially conducted within the German collaborative research center "SFB Humanoid Robots" granted by Deutsche Forschungsgemeinschaft and within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion").

References

[1] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner, and M. Bordegoni, "Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm," in *Proc. 9th ISRR*, Snowbird, UT, Oct. 9–12, 1999, pp. 229–238.

[2] S. Schaal, "Is imitation learning the rout to humanoid robots?" *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, Jun. 1999.

[3] M. Kaiser and R. Dillmann, "Hierarchical learning of efficient skill application for autonomous robots," in *Proc. Int. Symp. Intell. Robot. Syst.*, 1995, pp. 28–33. [Online]. Available: citeseer.ifi.unizh.ch/kaiser95hierarchical.html
[4] C. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. 14th ICML*, D. H. Fisher Jr., Ed., 1997, pp. 12–20.
[5] S. Calinon and A. Billard, *Learning of Gestures by Imitation in a Humanoid Robot*. Cambridge, U.K.: Cambridge Univ. Press, 2005. In Press.
[6] A. Alissandriakis, C. Nehaniv, K. Dautenhahn, and J. Saunders, "Using Jabberwocky to achieve corresponding effects: Imitating in context across multiple platforms," in *Proc. Workshop Social Mechanisms Robot Program. Demonstration, ICRA*, 2005, pp. 9–12.
[7] Y. Sato, K. Bernardin, H. Kimura, and K. Ikeuchi, "Task analysis based on observing hands and objects by vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, 2002, pp. 1208–1213.
[8] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Trans. Robot. Autom.*, vol. 10, no. 6, pp. 799–822, Dec. 1994.
[9] J. Aleotti, S. Caselli, and M. Reggiani, "Toward programming of assembly tasks by demonstration in virtual environments," in *Proc. Int. Workshop Robot and Human Interactive Commun.*, Millbrae, CA, Oct. 31–Nov. 2, 2003, pp. 309–314.
[10] R. Zoellner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," in *Proc. ICRA*, Barcelona, Spain, 2005, pp. 1535–1540.
[11] J. Chen and A. Zelinsky, "Programming by demonstration: Coping with suboptimal teaching actions," *Int. J. Rob. Res.*, vol. 22, no. 5, pp. 299–319, May 2003.
[12] M. Nicolescu and M. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proc. 2nd Int. Joint Conf. Auton. Agents and Multiagent Syst.*, 2004, pp. 241–248.
[13] R. S. Sutton, D. Precup, and S. P. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, no. 1/2, pp. 181–211, Aug. 1999. [Online]. Available: citeseer.ist.psu.edu/sutton99between.html
[14] D. Newtson, G. Engquist, and J. Bois, "The objective basis of behavior units," *J. Pers. Soc. Psychol.*, vol. 35, no. 12, pp. 847–862, 1977.
[15] M. Ehrenmann, R. Zöllner, O. Rogalla, and R. Dillmann, "Programming service tasks in household environments by human demonstration," in *Proc. 11th IEEE Int. Workshop Robot and Human Interactive Commun. (ROMAN)*, Berlin, Germany, Sep. 25–27, 2002, pp. 460–467.
[16] R. Zoellner and R. Dillmann, "Using multiple probabilistic hypothesis for programming one and two hand manipulation by demonstration," in *Proc. IEEE/RSJ Int. Conf. IROS*, Las Vegas, NV, Oct. 2003, pp. 2926–2931.
[17] H. Friedrich, V. Grossmann, M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann, "Towards cognitive elementary operators: Grasp classification using neural network classifiers," in *Proc. IASTED Int. Conf. ISC*, Santa Barbara, CA, Oct. 1999, vol. 1, pp. 121–126.
[18] K. Erol, J. Hendler, and D. S. Nauin "Semantics for HTN planning," Univ. Maryland, College Park, MD, Tech. Rep. CS-TR-3239, 1994. [Online]. Available: citeseer.ist.psu.edu/erol94semantics.html
[19] R. M. G. DeJong, "Explanation-based learning—An alternative view," in *Mach. Learn.*, vol. 1, 1986, pp. 145–176.
[20] T. M. Mitchell, "Generalization as search," *Artif. Intell.*, vol. 18, no. 2, pp. 203–226, 1982.
[21] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in *Proc IEEE/RSJ Int. Conf. IROS*, 2004, pp. 2769–2774.
[22] R. G. P. Smyth, "Rule induction using information theory," in *Knowledge Discovery in Databases*. Cambridge, MA: MIT Press, 1991, pp. 159–176.

Michael Pardowitz received the Master's degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 2004, where he is currently working toward the Ph.D. degree.

His interests include machine learning, human–robot interaction, and task learning for service robots.

Steffen Knoop received the Master's degree in electrical engineering and information technology from the University of Karlsruhe, Karlsruhe, Germany, in 2003, where he is currently working toward the Ph.D. degree.

His interests include human–robot interaction, as well as task description and execution for mobile service robots.

Ruediger Dillmann (M'93) received the Ph.D. degree in electrical engineering from the University of Karlsruhe, Karlsruhe, Germany, in 1980.

Since 1987, he has been a Professor with the Department of Computer Science and Technical Engineering, and since 2001, he has been Director of the research group Industrial Applications of Informatics and Microsystems with the University of Karlsruhe. Since 2002, he has also been Director of the Research Centre for Information Science at the University of Karlsruhe. He is involved in many synergistic activities: Director of the German collaborative research center "Humanoid Robots," IEEE Robotics and Automation Society chairman of the German chapter, Chairman of the German Society of Information Science, section 4.3/1.4 "Robotic Systems," Chairman of the German Association of Engineers, Editor in chief for the journal *Robotics and Autonomous Systems*.

Raoul D. Zöllner received the Ph.D. degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 2005. The focus of his thesis was on learning bimanual manipulation tasks from human demonstration.

In May 2006, he joined the research center "Information and Communications" at Siemens AG, Corporate Technology, Munich, Germany. The main research interests are intelligent, autonomous systems, especially field perception and understanding, machine learning, and decision making.