FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D.4.3.1
Report on selected categorizations of innate skills, skill representation and implementation. Evaluation of machine learning techniques for enabling one-shot learning.

Due date of deliverable: 31.12.04
Actual submission date: 31.01.05

Start date of project: January 1st, 2004 **Duration** : 48 months

Organisation name of lead contractor for this deliverable:
University of Karlsruhe

Revision: final
Dissemination level: PU

Executive Summary

Learning and acquiring task knowledge represents a major property of a cognitive robot companion and in the Research Activity (RA) 4 of the COGNIRON project the scientific challenges of skill and task learning are addressed. This report describes the work conducted by the University of Karlsruhe within the WP4.3 during the fist phase of the project. The aim of this workpackage was to set up scientific base for learning complex tasks and especially to investigate the role of background knowledge for the leaning process. The implementation of the work has been carried out within the framework of the Programming by Demonstration System developed at the UKA.

The chosen approach within this work package is investigating how learning and knowledge acquisition based on very few or even only one (positive) example can be used for task learning. Like humans a robot is supposed to learn or to understand new facts by watching or interacting with humans intuitively and this means that the learning process must be incremental and avoid multiple examples of the same task. The success of such learning methods depends on the prior knowledge and on the reusability of learnt or acquired knowledge of the system. In order to reduce the problem complexity the work done in the first phase of the project focuses on manipulation tasks only. For the household domain and especially constrained on manipulation a general model for task decomposition according to a classification of manipulation task was created and implemented. Based on this, methods for extracting, classifying and analysing information from a human demonstration were implemented and evaluated. The applied learning and knowledge acquisition strategies are derived from the state of the art and the knowledge analysis, which is done by humans in order to understand and reason about manipulation tasks in household environments.

Role of "Report on selected categorizations of innate skills, skill representation and implementation. Evaluation of machine learning techniques for enabling one-shot learning" in Cogniron

As stated above, learning and knowledge acquisition are crucial for a robot companion, who is supposed to coexist with humans. The study of learning tasks from humans by watching them and interacting with them, without the need of special training sets is one step towards scalable learning systems and lifelong learning. Therefore the work carried out so far in WP4.3 builds a strong base for further investigating the learning of and reasoning over complex task in a household environment. The used and acquired knowledge base on complex task knowledge is strongly liked to the work done in the WP4.1 ("*Sub-goals extraction and metrics of imitation performance*") and WP4.2 ("*Corresponding mapping across dissimilar bodies*"). The methods for single-shot learning of complex tasks implemented in WP4.3 ("*Prior Knowledge and data representation for Imitation learning; Algorithms for one-shot learning*") employ results from WP4.1 to identify the key points and task goals of a demonstration sequence that guide the later interpretation with the methods described in this report. Further, for transferring learned or acquired problem solving strategies though different embodiments (investigated by WP4.2) special knowledge has to be extracted and integrated in the task description.

Further the work has a great impact for the research done in RA 6 ("*Intentionality and Initiative*") since acquired or learned task knowledge has to be executed through the robot according to the adequate situation and context. Therefore the task representation is done in close collaboration with RA6 as well as the required specification of information for their execution.

In order to learn intuitively from a human user demonstrating a task he or she has to be observed and his or her activities have to be interpreted in a way suitable for extracting task knowledge. Hence the results from RA2 (*"Detection and understanding of human activity"*) are highly relevant for and integrated in the work of WP4.3.

Relation to the Key Experiments

The results of this work package will be integrated in the KE 3.3 "Skill and Task Learning" and especially will be included in the Cogniron Function CF-LCT (Learning complex task descriptions from elementary operations).
For KE 3 this work package provides the robot the abilities to learn and reason over the reestablishment of spatial relations between objects as well as their functional rolls with a certain task as they were demonstrated by a human teacher in a training session. The results will be demonstrated in the second script of KE 3. This demonstration will consist of observing a user laying the table respectively carrying out context dependent "fetch and carry tasks" in the context of serving guests. The results obtained from work on learning object features and descriptions will be integrated to learn completely new tasks.
The results from WP4.3 are directly contributing to CF-LCT by rooting the functionality in scientific insights. They provide methods for modelling complex tasks that appear in Key Experiment 3, Script 2. The developed framework for learning complex task descriptions is employed in CF-LCT to model, perceive and record task demonstrations and tasks originating from the household domain. This enables the Robot Companion to adapt in a flexible and intuitive way to the individual and diversifying environs and needs of the robot's user.

1 Report on selected categorizations of innate skills, skill representation and implementation. Evaluation of machine learning techniques for enabling one-shot learning.

Introduction

During the last couple of years, Human-Centered Robotics (HCR) became a major trend in the robotics community. One crucial point in HCR is to build extensible systems that adapt to unstructured situations and new tasks that the human (user) comes up with. This can be achieved by equipping robots with learning capabilities that facilitate the transformation of situational and task knowledge a user has into procedural knowledge utilizable by a robot companion. Humans usually do not have the patience to repeat a single task demonstration several hundred or even more times to provide inductive learning systems with a sufficient amount of training examples. Therefore different approaches that content with only a few or even a single positive example have to be taken into account for human-centered learning.

A human performance of any complex task to be reproduced by a Robot Companion is based on several innate skills that are the syntactical foundation of every composite task. The objectives of WP 4.3 in the first project phase were to identify those elementary skills and appropriate representations for them as well as models for the decomposition of overall tasks into sets of applied skills. Furthermore, work on the background knowledge mandatory for describing the semantics of each skill, like its applicability constraints and the achieved results of its application, is required. Last, existing paradigms of machine learning have been analysed with respect to their appropriateness for learning combinations of skills from sequential demonstrations. This assay attaches special importance to the amount of training data and the background knowledge base required for learning a task. A Robot Companion is much more likely to be accepted by the user if it can learn complex tasks from very few or even a single task demonstration (one-shot learning).

This document starts with the presentation of different learning paradigms and their appropriateness for one shot learning followed by a classification of knowledge which can be used for these learning strategies. Focusing on manipulation tasks a general hierarchical model and a classification of manipulation tasks is discussed and it is pointed out that the hierarchical representation of tasks improves the explicability and the scalability of the learning process. Further the role of basic (innate) skills representing the lowest level (hardware abstraction layer) of the hierarchical task description during the knowledge acquisition process and the execution of task is outlined. Finally, the task representation, the learning strategy and a short example of the first implementation is shown.

State of the Art

Several programming systems and approaches based on human demonstrations have been proposed during the past years. Many of them address special problems or a special subset of objects only. An overview and classification of the approaches can be found in [5] [6]. Basis for the mapping of a demonstration to a robot system are the task representation and task analysis. Often, the analysis of a demonstration takes place observing the changes in the scene, described by using relational expressions or contact relations [7] [8].

Issues for learning to map action sequences to dissimilar agents have been investigated by [9]. Here, the agent learns an explicit correspondence between his own possible actions and the actions performed by a demonstrator agent by imitation. In [10] the authors concentrate on the role of interaction during task learning. They use multiple demonstrations to teach a single task. After generalization they use the teacher's feedback to refine the task knowledge.

To generalize a single demonstration mainly explanation based methods are used [11] [12]. They allow for an adequate generalization taken from only one example (One-Shot-Learning).

A generalized task description that is to be executed must be mapped onto an executable robot program. Several robot task representations have been proposed in the past. They can be roughly classified into programming languages including control structures and declarative sequence- or tree-like descriptions of the task. An overview can be found in [13].

Learning paradigms and their appropriateness to one-shot learning

Many learning systems, especially ones that utilize inductive or statistical approaches, require a large amount of training data in order to generalize the demonstrated examples as far as possible. This generalization allows for an extension to unknown instances and application to problems not contained in the training set. Such systems lack the ability to significantly learn from a few training instances and transfer the knowledge acquired during this process to different new instances not contained in the training samples. In contrast to this, learning humans clearly demonstrate their capability to learn from single demonstrations and tend to expect that from a robot companion, too. One possible explanation for the superiority of human learning is that they exploit earlier experiences and extensively employ prior knowledge. From this considerations follows, that in addition to the established learning approaches the influence of methods incorporating prior knowledge aiming towards one-shot learning has to be analyzed and evaluated.

According to [1] [2] four major paradigms of machine learning can be identified: Inductive learning methods build a description of the entity to be learnt which covers or describes all positive training instances and none of the counterexamples. Analytic methods utilize solved problems (the examples) to aid deductive reasoning on a formalized or axiomatic domain theory in order to improve system performance on certain fields rather than learning completely new ones. These two approaches form the category of symbolic machine learning, as opposed to subsymbolic machine learning. The latter contains connectionist methods, which view learning as the problem of assigning and adjusting weights to elements of a network structure that guides distributed information processing, affecting the outcome in order to converge to the target function. Last, there is the genetic paradigm that relies on parallel mutation, evaluation and selection of possible target concepts to ensure their improvement.

In order to evaluate these approaches with regard to one-shot task learning in household scenarios the prerequisites and requirements of the occurring learning jobs have to be further specified. In the following, this documents deals with learning of tasks in the household domain. These tasks include fetch-and-carry operations, transports of objects, operating domestic appliances, usage of tools and optimizing manipulations like handing over objects from one hand to the other. A system for one-shot task learning should be capable of observing the human user performing a task and detecting and segmenting the single actions that form the demonstrated task. Further on it requires the ability to interpret the demonstration with regard to the influence it has on the environment and abstract it as far as possible to different environments and similar tasks. After the learning process is complete, the task is ready to be mapped on specific hardware architecture and can be executed by a robot in the environment. The data representation that describes a task should aid this process by providing the learning system all necessary sensorial information in an appropriate manner and has to be flexible enough to model all possible actions stated above in a way that allows to describe them as generally as possible.

Further restrictions arise from the application domain: Out of safety reasons it is mandatory to have methods to verify the correctness of learned data. Learning methods should be performed online in order to enable user-friendly interfaces. The user should interact with the learning system in a way that allows him to monitor the hypothesis the system makes and correct them if necessary. The task representation and the learning algorithms should be independent of the specific robot hardware because this allows executing learned tasks on different hardware platforms. Learning should be incremental and not batch-style, giving the user the possibility to add new tasks whenever they are needed and not in an overlong and tedious initial teaching phase.

In [3] it is suggested that symbolic learning approaches could be a little more suitable for those kinds of learning problems than subsymbolic ones, especially connectionist paradigms. This statement can be sustained by the fact that many symbolic learning approaches seem to outperform subsymbolic paradigms in terms of classification correctness and learning rate given relatively small data sets. In the earlier defined domain of task learning, these training sets can contain very few (less than five) training samples or even only a single task demonstration. So, symbolic paradigms seem to be more promising in this specific domain. Additionally the learned hypothesis a symbolic learning algorithm outputs are usually easier to interpret, especially by human. This is likely to lead to a more effective and easier verification of the learned results. Therefore safety issues can be tackled in a direct and more reliable manner. Last, many symbolic learning techniques do not require batch processing, which means they feature incremental learning. This enables the system to learn a task even from a single user demonstration and later refine the learned knowledge when the user has the time and feels like providing another one ore more demonstration.

Knowledge base for complex tasks

As stated above learning from few examples is strongly related to an adequate associated knowledge base, which covers the domain knowledge. The management of the knowledge base should support the integration of new acquired knowledge, so that it can be used for further learning cycles.

The following classification of knowledge can be made:

- Environmental knowledge (objects, scene model etc.)
  For understanding a given demonstration of a task a representation of the environment including descriptions and features of parts of the environments like objects is necessary. The environmental description should include a scene and object model as well as methods for detecting and specifying changes in it.

- Functional knowledge (roles of objects, functional description of tools etc)
  Looking at manipulation tasks the goal extraction of a demonstrated action is strongly related to the functional features of the manipulated object. For tool handling tasks in general the role of the tool and the type of interaction with the manipulated object is sufficient to determine the functional goal of the demonstration.

- Temporal / spatial knowledge (move types, synchronisation etc)
  The description and detection of environmental changes over the time is essential for understanding the movements during a task execution. Herby the classification of move types in general and of grasps in the case of manipulation tasks is one feasible way for describing a and detecting movements. For understanding complex tasks which are executed bimanually a synchronisation of the hand movements is required and thus methods for finding key points for synchronisation have to be integrated.

- Commonsense knowledge (physics, gravity force, stability criteria etc.)
  For improving the understanding of a demonstrated task commonsense knowledge like physical constrains can be integrated. Based on these, automatic sensor data correction

methods can be applied in order to increase the reliability and robustness of learning process. Furthermore out of this knowledge plausibility criteria can be defined in order to increase the quality of the analysis and interpretation process of an observed task.

- Special task knowledge (relevant / special features, criteria etc.)
Apart from general knowledge presented above, special knowledge, which is valid for only one or a few peculiar tasks, can be necessary in some cases for goal detection of an observed task. This kind of knowledge is very hard specify and cased based reasoning combined with methods used in expert systems seam feasible for acquiring and exploiting it.

Models of manipulation tasks for household domains

In the framework of this work package manipulation tasks are defined as tasks during which an interaction between a robot gripper or a human hand and the environment respectively an object is performed. Further this interaction is modelled as a grasp, whereas touching an object i.e. while pushing a button denotes a specific (degenerated) grasp. This simplification enables a consistent modelling of manipulation tasks without restrictions.

Manipulation task performed intuitively by one human using both hands can be classified according the effects they have in the environment during and after the manipulation as follows:

1. Transport actions.
2. Device handling
3. Tool handling

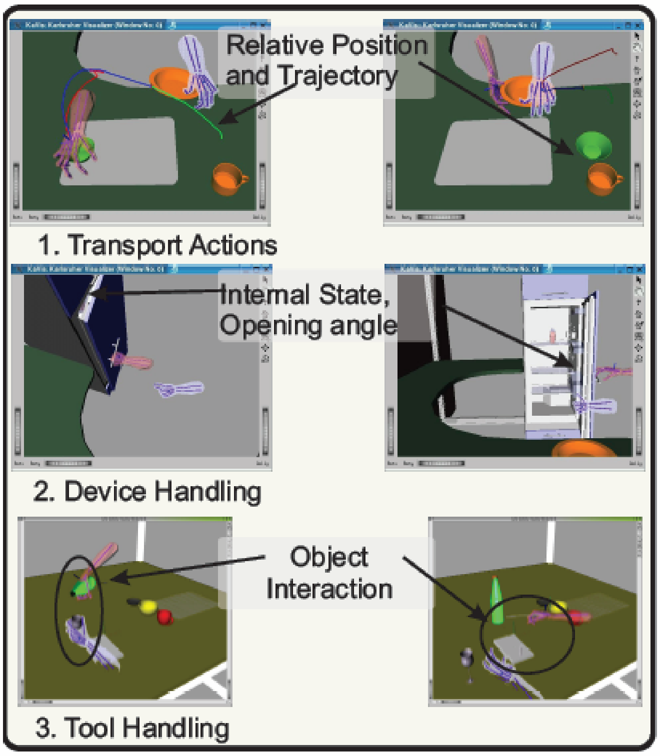

Fig. 1 : Manipulation Task Classes and their Relevant Effects.

First, there are transport operations like pick & place transports or fetch & carry tasks, where the focus lays on the transport of one or more objects (see Fig. 1). Hereby the Cartesian position or more general the spatial information over time builds the characteristic description parameter. Transport actions are a part of almost all manipulation tasks. Second, device handling like opening a drawer or operating the microwave oven are forming a other class of tasks, where in contrast to the first one the internal state of objects is manipulated. Last, tool handling actions like screwing or pouring in a glass of water are building the most complex manipulation class. In contrast to the other two classes, where the manipulation is performed by the human or robotic hand, here the manipulation is performed indirectly by means of a tool, which is manipulated directly. Therefore, the main parameter for characterizing this class is the interaction between objects during the manipulation.

Modelling the distinguished manipulation classes requires a hierarchical approach, in order to increase both the flexibility and the saleability of task representation. Further a hierarchical representation is closer to the way humans would decompose a task and therefore supports the exchange of information between the robot companion and humans. The lowest level of the task model is build by a sequence of primitives called elementary operators (EO's). The EO's are denoting basic (innate) skills, which are characterized by a tight sensory motor coupling. Examples of elementary operation are move types or patterns like linear of spline moves or pouring actions.

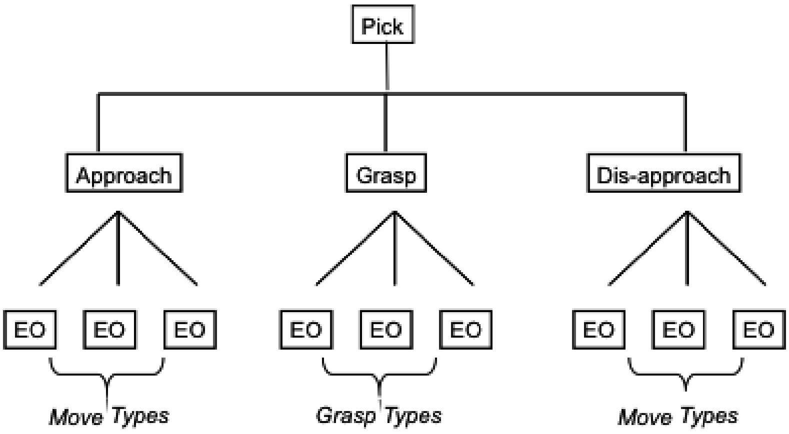

Fig. 2 : Hierarchical Model of a Pick Operation

The basic assumption made is that every sample from any manipulation class described above consists of at least a grasp and an ungrasp action. Pushing an object or touching a button during the operation of domestic devices can easily be mapped on grap / ungrasp actions. When an object is grasped, this constitutes a pick operation that consists of three parts: an approach movement, the actual grasping and a depart movement (see Fig. 2). Each of these sub-parts consists of an sequence of elementary operators. The place operations are modelled analogously.

In between a pick and place operation, depending on the manipulation class, several basic manipulation operations can be placed (Fig. 3). E.g. a demonstration of the task "'puring a glass of water"' consists the basic operations: "'pick a bottle"' "'transport the bottle"', "'pure 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.

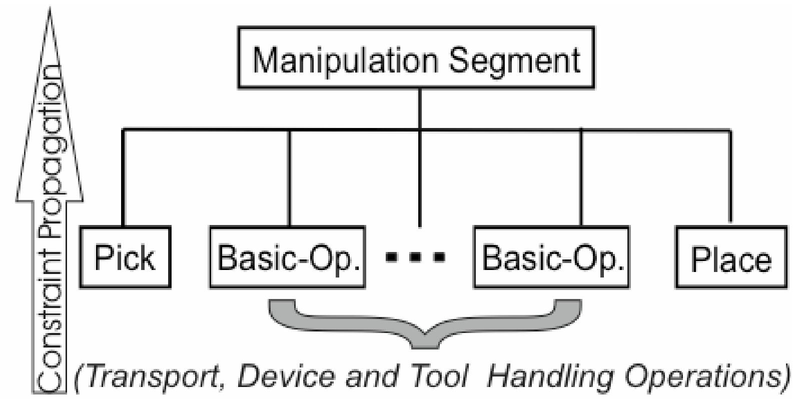

Fig. 3 : Representation of Manipulation Segments

The level of manipulation segments denotes a new abstraction level on closed sub tasks of manipulation. In this context closed means that, a manipulation segment ensures that both hands are free and that the environmental state is stable. Furthermore the synchronization of EO's for left and right hands are included in the manipulation segments. Pre and post conditions 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.

Innate skills for manipulation tasks

Innate skills denote a set of predefined skills of the robot serving for both: the execution of manipulations and the learning of new tasks. Hereby the requirements and the instances of the two peculiarities differ depending on the dissimilarity between the embodiment of the robot and humans. According to the models presented in the last section, the EO's consisting of basic grasp and move types could be the minimum set of innate skills for manipulation tasks. Never the less for interpreting human demonstrations and in particular for the segmentation of the recorded sensory data of the demonstrated task more information than move and grasp types is needed. Table 1 shows a selection of the main parameters, which are needed for a robust segmentation. However as stated above the more information out of the background knowledge is integrated in the system the more reliable results can be achieved.

Basic Skill	Parameter
Grasp: Static dynamic	Hand: TCP velocity, joint angles +forces
Move types	TCP trajectory analysis
Device handling : Open doors drawers Push/rotation buttons	+Object model (Type) Move-axis, handle, state Ditto
Tool handling: Screwing Pouring	+Functional role "screw-able" "pour in / out"

Table 1 : Parameters for Basic Skills

The execution of complex tasks means the sequential execution of innate skills respectively learned skills. Generally, all manipulation tasks can be described with a set of move and grasp types, but than there is a need for a high amount of control parameters for triggering the single skills and hence the reliability of the execution is limited. Therefore, especially for tool handling tasks, it might be appropriate to uses some high level skills. For example, the task "pouring a glass of water" can be described by only grasp and move types, but in this case the loop of the pouring process must be controlled by the task description. Since pouring fluids is a common task in household domains it is appropriate to implement a special innate skill for it. This will incorporate the pouring loop and will be parameterized with the two objects (the glass and the bottle) and the desired fluid level.

Due to the different embodiments the execution of a complex task learned form humans can not necessary be exactly performed like humans would do. But the effect of the task in the environment has to be achieved. The abstract representation of tasks should incorporate as much as possible information for enabling the mapping between embodiments. Further the aim of the work of this work package is to find a general abstract robot invariant task description, which does not depend on a certain embodiment of a robot.

Applied learning and knowledge acquisition strategies

In order to build the data structures described in the preceding chapter from a user demonstration several steps using different learning and knowledge acquisition methods are applied in order to build the hierarchical task representation in a bottom-up strategy. On the lowest level, the sensor abstraction layer, at every timeframe the sensor data is recorded and preprocessed. Classifiers are applied that transform the subsymbolic sensor values into symbolic information, according to the elementary operators. On later stages, this sequence of symbols is analytically transformed, applying syntactical and semantically rules, to form a generalized task description that encodes an interpretation of the learned task. The single computation steps are further explained in this section.

In a first step the static grasps are classified according to the hierarchy proposed by Cutkosky [4] and the dynamic grasps according to the classification proposed by Zöllner [14]. For this Neuronal Networks and the Support Vector Machines were set up and trained. The classification of the correct grasp type enables the user to give hints to a robot in an execution environment in order to ease the instantiation of a learned task. The moving operations in the user demonstrations that are located between the grasp and release operations have to be fragmented into motion segments that directly correspond to basic movement types a robot can execute. The movement types used in the system are linear, point-to-point and spline moves. Rule sets based on the geometrical features of the trajectories were used to split the whole trajectory into pieces and classify these pieces according to the movement primitives.

Grasp classification and the fragmentation of movement primitives allows the transition from sub-symbolic sensory data to semantic symbols and features that are applicable to task-oriented reasoning. At this stage, a task <T> is represented by a sequence of elementary operators $<EO_i>$:

$$< T >=< EO_1 >< EO_2 >< EO_3 > \ldots < EO_n >$$

Starting from this symbolic information the hierarchic structure of the task tree is built.

In the next step the trajectory segments are accumulated to approach, disapproach and basic operators. In the following only the class of transport tasks is considered, were the basic operators are exclusively transport movements. For this, the contexts of a single movement primitive are analysed and at the points where it changes so called “context-switches” are introduced. Context-switches mark a transition between approach, depart or transport movements. They are determined using a metric on the distance to the grasp and ungrasp points combined with the velocity of the hand TCP and the hand pose. Once these context switches are established, the basic movement primitives are chained together to form the approach, depart and transport trajectories. The resulting task description is a sequence of approach (A), disapproach (D), transport (T), grasping (G) and ungrasping(U) operations:

$$< T >=< A_1 >< G_1 >< D_1 >< T_1 >< A_2 >< U_1 >< D_2 > \ldots < A_n >< U_n >< D_n >$$

On the next level, the pick and the place operations, consisting of an approach and a depart trajectory and a gasp or ungrasp action, respectively, are formed. The representation of a pick/place operation is as follows:

$$< Pick >=< A >< G >< D > \text{ resp. } < Place >=< A >< U >< D >$$

These are the basic building blocks that carry semantic information with respect to the goals of a user demonstration. Every such an operation causes changes in the spatial arrangement of objects in the scene. This spatial arrangement, as a part of the state, is expressed in terms of spatial relations between the objects in the scene. Every pick and every place action changes these relations and the according state. A pick operation deletes all relations between the picked entity and every other object in the environment and a place operation re-establishes a different set of relations, depending on the position the object is placed on. These changes are computed and represented using lists of relations. Changes are represented as a set of relations between two objects and their value (true or false) and a function CH is established that assigns the changes to every pick operation:

$$CH(< pick_i >) = \bigcup_j (rel_j obj1_j obj2_j val_j)$$

The CH-function is defined analogously for place operations.

Combining a pick- and a place-operation to a pick&place-operation causes the computation of the overall changes carried out by this operation in the scene. The changes of each sub-part are merged to a list of effects that this pick&place operation has. Formally, a pick&place-operation is defined as

$$< P \& P_i > = < Pick_i >< Transport_j >^* < Place_i >.$$

The changes for the Pick&Place-operation are calculated as

$$CH(< P \& P_i >) = CH(< Pick_i >) \bigcup CH(< Place_i >) .$$

Once a sequence of pick&place-operators are extracted from the user demonstration, the overall goals of the task can be deduced. This is done by sequentially accounting all the changes performed by any pick&place-operation. Unnecessary changes that are invalidated by any later action are detected and deleted from the set of changes:

$$CH(< P \& P_i >) = CH(< P \& P_i >) \backslash \{ \bigcup_{j>i} CH(< P \& P_j >) \}$$

When any set of changes becomes the empty set, this means that every change induced by the according operation is deleted by any later action and the according operation does not contribute to the goals of the sequence and should be pruned. So only the operations that move the state of the environment from the initial towards the goal state are regarded in the following steps and information from the unnecessary operations are discarded.

The result of these steps is a hierarchical task description including its goals and subgoals that is built from a single user demonstration, consisting only of the necessary actions and leaving out the operations demonstrated unintentionally by the user. This hierarchical task representation is further generalized beyond the single user demonstration in order to be applicable in similar environments. This generalization is not data-driven because single-shot learning cannot provide the necessary large data sets needed for data-driven generalization. Instead of that background knowledge in form of rules is applied that generalizes a single example over space (trajectories are classified based on the accuracy needed during the execution time) and object classes (generalization from a specific object to the class of objects it belongs to).

Additionally, a semantic analysis of the scene has to be done. For this, statistics are maintained for object co-occurrences. Whenever special objects are used, like tools or containers that can perform several specific operations (pouring liquids from a bottle into a glass, operating a screwdriver), these statistics are evaluated to obtain information about which operation is most likely in the current context and which object is most likely to be referenced by this operation. This process is called statistical generalization.

Hierarchical Task Representations – Macro Operators

Once the user demonstration is analysed, the compiled information is enriched with structural information describing the attributes of the objects present at the scene and execution constraints. The result is mapped onto an abstract structure that combines all available knowledge into a representation suitable for execution. This is formed similar to the representation chosen in the STRIPS system, called macro-operators. Every macro-operator is associated with its signature, containing the total pre- and post conditions, in order to enable plugging together macro-operators that happen to have matching signatures.

Each operation in a segment-tree analyzed in the way described in the preceding chapter encapsulates certain pre- and post-conditions that describe the state of the environment at the beginning and the end of that certain operation. These conditions are propagated from the elementary operations up to the manipulation segment level. A generalization regarding the position and object types and their features is done in parallel. The total pre- and post-conditions of all manipulation segments of a macro-operator are propagated to a context, under which the macro-operator is executable, and to effects that an execution of the macro-operator has on the environment. A positive evaluation of the context enables its execution and an error free execution leads to the desired effects in the environment.

Several different user demonstrations either performed in sequence one after the other or with interruption (the more usual way an end user builds the task knowledge of a robot companion) each form a different macro operator that is added to a knowledge base of learned and executable tasks. Execution of a macro-operator is done by instantiating the generalized objects with objects present in the execution environment. Macro-operators can be selected from this database for a specific environment and a specific job to be accomplished. This selection is guided by the according pre- and post-conditions that the execution presupposes and provokes respectively. The topic of multiple macro-operators applicable in a certain situation is not dealt with in the current project phase.

Experimental Results

In this section an example task and the analysed results are described in order to show the functionality of the implemented system.

The initial state of the environment is shown in Fig. 4. The objects apparent in the scene are a desk with a silver tray, a plate, a small bowl and a cup. In the demonstration, first the left hand picks the plate. In parallel, the right hand picks the bowl. Next the plate is placed on the silver tray and afterwards the bowl is placed on the plate.

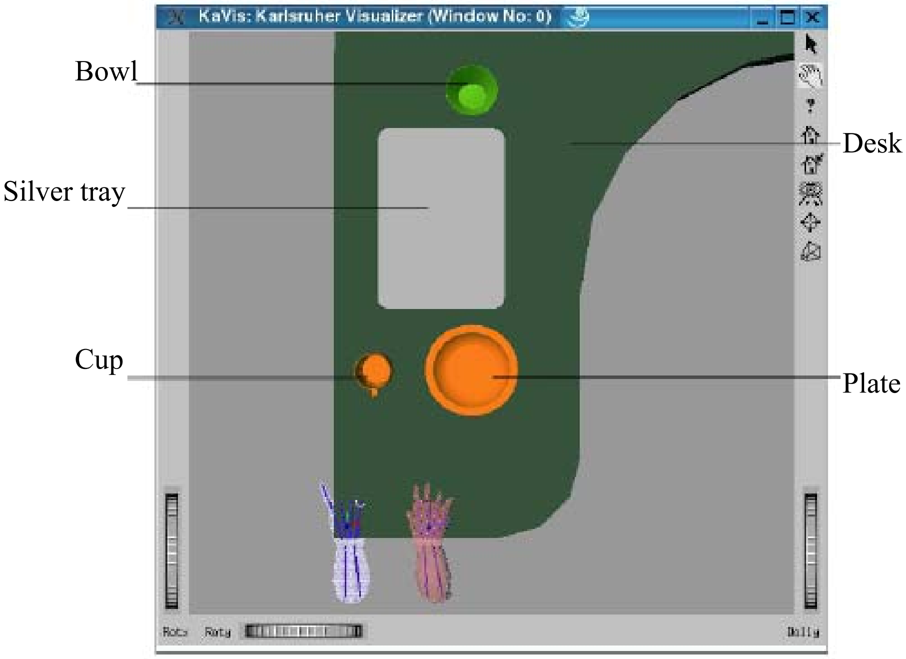

Fig. 4 : Initial environmental state of the described task demonstration

The analysis of the scene detects the following actions:

1. The grasp operation of the plate (Fig. 5)
2. Grasping the bowl (Fig. 6)
3. The ungrasp operation of the plate (Fig. 7)
4. Releasing the bowl (Fig. 8)

Fig. 6 : Grasping the plate

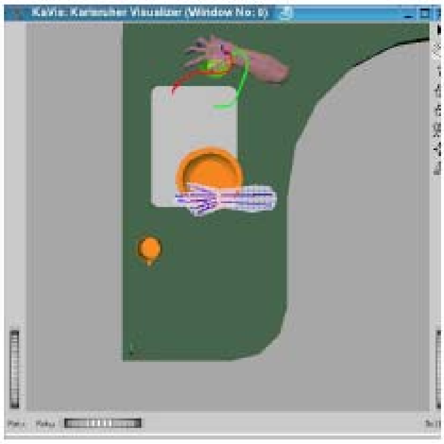

Fig. 5 : Grasping the bowl

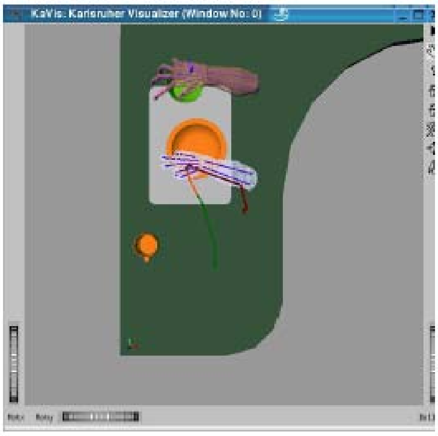

Fig. 8 : Ungrasping the plate

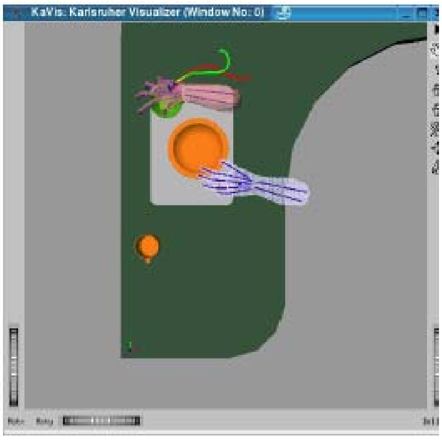

Fig. 7 : Ungrasping the bowl

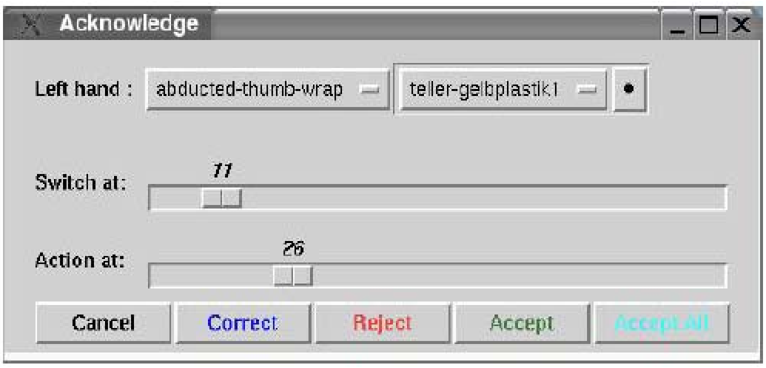

Fig. 9 : Hypothesis confirmation dialogue

At each step the systems requires the user to either confirm the learnt hypothesis and accept the detected operation or, alternatively, correct or reject the operation (Fig. 9). This is useful to add the possibility to correct errors occurring from the transition from the subsymbolic sensor data to the symbolic entities that form the semantics of a task.

Fig. 10 : Resulting Segmentation

The segmentation of this task, performed with the methods described in the preceeding sections and the result achieved is visualized in Fig. 10. The resulting macro-operator, that is added to the knowledge base of stored macro-operators, together with it's context and contributions is presented in Fig. 11.

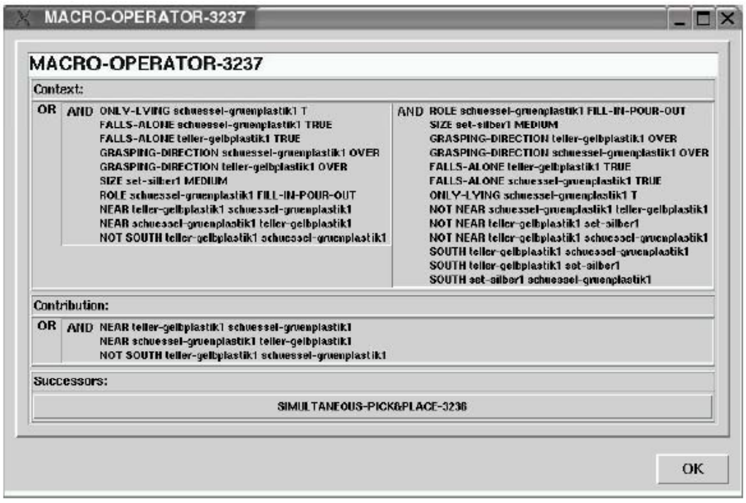

Fig. 11 : Presentation of the macro-operator to the user

The example shows how user demonstrations can be analysed in the proposed way and task descriptions can be successfully generated from a single user demonstration and can be generalized to be applicable in significantly different but similar in terms of the task to be done. In particular, the system successfully executes the following of operations:

- Automatically detection and segmentation of two-handed manipulations according to the modelled classes
- Recognition of spatial, temporal and object-based dependencies of both hands
- Detection of relevant actions and pruning of unnecessary operations is successfully performed

2 Future Work

As mentioned in the last section, there can be multiple macro-operators applicable in a certain situation, resulting from multiple user demonstrations of the same task. The issue of dealing with these duplications of task knowledge and exploiting it as far as possible will be tackled in the next project phase. This will include learning task attributes that cannot be learned from a single example like reordering possibilities or pruning of unnecessary steps.

3 References

3.1 Applicable documents

Other related project documents (i.e., other deliverables) D.4.1.1, D.4.2.1

3.2 References

[1] J. G. Carbonell : Paradigms for machine learning. Machine Learning: Paradigms and Methods, Carbonell, J. G. (Ed.), MIT Press/ Elsevier, 1990.

[2] G. Briscoe, T. Caelli: A compendium of machine learning, vol.1. Ablex Publishing, 1996.

[3] J. Wnek, R. Michalski: Experimental comparision of symbolic and subsymbolic learning, HEURISTICS, The Journal of Knowledge Engineering, Special Issue on Knowledge Acquisition and Machine Learning, Vol. 5, No. 4, pp. 1-21, 1992.

[4] M.R. Cutkosky: On grasp choice, grasp models and the design of hands for manufacturing tasks. IEEE Trans. Robotics and Automation, 5(3):269-279, 1989.

[5] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner and M. Bordegoni, "Learning Robot Behaviour and Skills basd on Human Demonstration and Advice: The Machine Learning Paradigm", in: 9th Intl. Symposium of Robotics Research (ISSR 1999), Snowbird, Utah, 9.-12. Oct, 1999, pp. 229-238.

[6] S. Schaal, "Is imiation learning the route to humanoid robots?" in Trends in Cognitive Sciences, vol. 3, 1999, pp. 223-242.

[7] Y. Kuniyoshi, M. Inaba and H. Inoue, "Learning by watching: Extracting Reusable Task knowledge from Visual Observation of Human Performance", IEEE transactions on Robotics and Automation, vol. 10, no. 6, pp799-822, 1994.

[8] H. Onda, H. Hirukawa, F. Tomita, T. Suehiro and K. Takase, "Assembly Motion Teaching System using Position/Force Simulator – Generating Control Program", in 10th IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems (IROS), Grenoble, 1997, pp. 389-396.

[9] A. Alissandrakis, C. Nehaniv and K. Dautenhahn, "Imitation with ALICE: Learning to Imitate Corresponding Actions Across Dissimilar Embodiments", IEEE Trans. Systems, Man and Cybernetics, vol. 32, no. 4, pp. 482-496, 2002.

[10] M. Nicolescu and M. Mataric, "Natural methods for robot task learning: instructive demonstrations, generalization and practice", in: 2nd Intl. Joint Conf. On Autonomous agents and multiagent systems, 2003, Melbourne, Australia, pp 241-248, 2003.

[11] T. Mitchell, "Explanation-based generalization – a unifying view", Machine Learning, vol. 1, pp 47-80, 1986.

[12] H. Friedrich, "Interaktive Programmierung von Manipulationssequenzen", Ph.D. Thesis, Universität Karlsruhe, 1998.

[13] I. Pembeci and G. Hager, "A comparative review of robot programming languages", CIRL Lab Technical Report, 2002

[14] R. Zöllner, O. Rogalla, R. Dillmann, M. Zöllner, "Understanding Users Intention: Programming Fine Manipulation Tasks by Demonstration", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002, Lausanne, Switzerland.