Human-Robot-Communication and Machine Learning

Abbr. title: Human-Robot-Communication and ML

Volker Klingspor
Universität Dortmund
Computer Science Dept. VIII
D-44221 Dortmund, Germany
Phone: +49 231 755 2499 Fax: +49 231 755 5105
email: volker@ls8.informatik.uni-dortmund.de

John Demiris
University of Edinburgh
Department of Artificial Intelligence
5 Forrest Hill
Edinburgh, UK
EH1 2QL
Phone: +44 131 650 3083 Fax: +44 131 650 6899
email: johnde@dai.ed.ac.uk

Michael Kaiser
Universität Karlsruhe
Institute for Real-Time Computer Systems and Robotics
Kaiserstr. 12
D-76128 Karlsruhe, Germany
Phone: +49 721 608 4051 Fax: +49 721 606 740
email: kaiser@ira.uka.de

Abstract

Human-Robot Interaction and especially Human-Robot Communication (HRC) is of primary importance for the development of robots that operate outside production lines and cooperate with humans. In this paper, we review the state of the art and discuss two complementary aspects of the role machine learning plays in HRC. First, we show how communication itself can benefit from learning, e.g. by building human-understandable symbols from a robot's perceptions and actions. Second, we investigate the power of non-verbal communication and imitation learning mechanisms for robot programming.

1 Introduction

Potential markets of enormous size exist for mobile robots in the areas of materials transport, mobile surveillance systems, and floor cleaning (Schraft, 1994). In addition, the idea of the "personal robot" or "personal robotic assistant" (e.g., for aiding the elderly or disabled) is lately receiving a lot of attention. Robot manufacturers all over the world are expecting to open a new, consumer-oriented market that will allow for sales figures far beyond those that can be obtained from today's industrial robots. However, to enable new robot applications with emphasis on service tasks, it is necessary to develop techniques which allow untrained users to make efficient and safe use of a robot.

Two basic aspects characterize the interaction between the robot system and the user (Fig. 1, (Dillmann et al., 1995)). Firstly, the user wants to configure and, if necessary, instruct the robot for the task at hand. Secondly, the user may want to supervise and monitor the robot's performance. In both cases, the user does not **operate** the robot. Instead, she wants to use it to accomplish some task. Therefore, the emphasis is on **what** the robot will do, not on **how** it should do it. As in Human-Computer Interaction (Card et al., 1983), designing interfaces for Human-Robot Communication* therefore involves the following aspects:

1. Firstly, the user must be provided with an interface that allows her to intuitively instruct the robot. Here, *instruction* involves to translate the user's intention into correct and executable robot programs.

2. Secondly, feedback must be provided to the user so that she can immediately understand what's happening on the robot's side. This task requires to translate internal, possibly low-level representations used by the robot into a representation that can be understood by the user.

In both cases, the user acts as a teacher who demonstrates her intention by explicitly performing the corresponding operation. The robot can be seen as an apprentice that learns from this demonstration. The learning tasks are therefore related to **learning from communication.** Furthermore, learning enables the user to provide the robot with instructions and it enables the robot to give feedback to the user. Therefore, learning improves both user

*Note that we use the term *Human-Robot Communication* (as opposed to *Human-Robot Interaction*) purposively, since we're concentrating on the communication aspects and do not, for example, consider **physical** interaction such as cooperative handling of workpieces. An interesting review of projects involving such interactions in a Human-Robot-Computer system can be found in (Anzai, 1993).

to robot and robot to user communication. This is **learning for communication**. It aims at improving the communication between the two agents, i.e., to learn communication skills.

Consequently, the need for enhancing Human-Robot Communication is closely related to the idea of allowing humans to **make use** of robots. Hence, we will treat the learning tasks from a rather objectivistic point of view, assuming that the reference for the meaning of symbols used for communication will be the user's understanding of these symbols. The robot does not – at least for communication purposes – construct its own symbols, but grounds the user-defined symbols onto its own perceptions and actions. This point of view is clearly in contrast to approaches in artificial life (AL) (Stewart, 1995). Complete autonomy – the main goal of research in AL (Steels, 1994) – and the emphasis on learning for survival (i.e., to enable the robot to wander around for several days without colliding with obstacles and to learn when it is necessary to re-charge the battery (Birk, 1996)) is less important here. In these approaches, the systems learn from the environment, instead of learning from communication with a user. Furthermore, learning results in an improvement of their behavior for a fixed purpose, instead of learning something which might be used to communicate with the user. However, we will consider robots that learn from each other via imitation as an additional mean to distribute information to and from a human user among several, possibly cooperating robots. In any case, our robots learn what we will call "skills", both for communication and for actual physical performance. Of course, other learning tasks can be supported by means of interaction. For example, (Asoh et al., 1996) use interaction between user and robot to acquire a map that is used for navigation.

In this paper, we will first analyze different approaches to Human-Robot Communication and the role learning plays in these approaches (Section 2). Afterwards, we will discuss the mutual dependence between learning and communication, both in Human-Robot Interaction and Robot-Robot Interaction. Furthermore, we will describe techniques to acquire knowledge **for** the communication and **from** the communication between a robot and a user. Two specific learning tasks, to learn a language common for user and robot (Section 3) and to skills from human demonstrations and by imitation of another robot (Section 4) will be discussed in detail. Finally, the lessons learned will be summarized.

2 Issues in and Examples of Human-Robot Communication

The usefulness of communication depends on the ability of the communicating entities – i.e., the human user and the robot – to understand each other.

This is especially true for the communication between teacher and pupil, since the pupil (usually the robot) performs self-modifications based on its understanding of the information obtained from the teacher. In HRC, the issues involved are:

Purpose of communication, i.e., what kind of information is exchanged during communication and what is the purpose of this information exchange. Here we also consider the **level of abstraction** on which the exchanged information is represented.

Communication media, i.e., how (e.g., verbally, via gestures, via explicit interaction) information is transferred.

Direction of communication, i.e., whether information flows from the user to the robot, vice versa, or in both directions.

In the following, we will discuss the three most important settings for Human-Robot Communication, namely communication for task specification and execution, communication for robot supervision, and communication for robot programming. In any of these cases, we'll pay particular attention to the basic issues identified earlier.

2.1 Communication for Task Specification and Execution

By **task specification,** we refer to the process of providing the robot with information about the task to be accomplished. In contrast to **programming,** task specification does not involve changes or extensions of the robot's action knowledge. Once a task has been specified, the robot performs both physical actions (e.g., actual motions) and cognitive actions (reasoning) to accomplish the task. Dependent on how much the robot knows about its environment and how skillful it is, further communication with the user might be necessary during the course of action.

An example of task specification via natural language is the system KANTRA (Längle et al., 1995). The user can tell the robot which action it should perform with which object, for example "Grasp the red screw driver". In this approach, perception and action are separated. Planning is done in a classical way, creating a detailed sequence of actions, to be broken down to the lowest level of action, and executed until some problem arises. This approach for planning yields a few problems (Agre and Chapman, 1990): real problems are computational intractable, planning is inadequate for worlds with unpredictable events, plans have to be very detailed, and plans do not relate to concrete situations.

Agre and Chapman view plans as one among several sources of information for deciding what to do next. They call their approach *plan as communication* (Agre and Chapman, 1990). Instead of defining a sequence of fixed and deterministic operators, plans just help to decide what's good to reach a given goal. Therefore, interpreting and executing a plan is more complex than in classical planning, because the interpreter itself must be able to solve the problem at least in a rudimentary way. To execute a complex sequence of actions, such a system must be accompanied by a user/instructor, and in each situation that requires a decision, new instructions must be given: "Head left now, ..., now rotate to the right, ..., slow down, ...". It cannot plan ahead to decide on what to do in the future in order to reach a goal.

The project "Situated artificial communicators", similarly to (Agre and Chapman, 1990), aims to embed commands into sequences of actions that are jointly perceived by the user and the system (Förster et al., 1995). The whole system integrates this planning and communication part with a natural language interface and an object identification system. Each command triggers a single behaviour like "grasping", and relates this behaviour to an object. Like above, the commands are not used to define complex goals.

For assembly robots generating a description of the reachable objects is relative easy, since all objects are visible. Mobile robots must invest more effort into generating a map of their environment through the need for extensive exploration. Whereas in Human-Human-Communication it is common to tell a person the way he should go, e.g. by saying "You go at the third crossing to the right, then at the second red light to the left, ...", this ability to interpret commands is currently not available for robots. Nevertheless, the idea is close to those of *plan as communication* (Agre and Chapman, 1990).

All the previously described approaches do not really combine knowledge for planning with knowledge for control. Either the system is reactive and thus has no planning capabilities. Or the system can communicate with the user, but operation execution and planning are separate and hence control is not reactive. Very few approaches combine reactive control with planning, e.g., (Saffiotti et al., 1995). These approaches, unfortunately, don't use learned concepts for communication. Learning, however, is in our opinion essential for building a common language for communication (see Section 3).

In addition to verbal communication, there exist many other ways to communicate with robots. The simplest way is to communicate via a joystick, but robots controlled in this way are not autonomous and the actions performed are rough. If the robot has a map of the environment, this map can be displayed on a touch screen, and the user can point to the position the robot should move to. This, however, requires a lot of knowledge about the environment, which is often not available. Even more knowledge is necessary, if the robot is controlled in virtual reality (Roßmann, 1995). Here the robot is controlled by a human seeing a virtual image of the environment the robot is working in. The human acts in this virtual world with a data-glove. This approach is often used to control a robot manipulator over a large distance, like in space applications, but the effort for constructing the virtual reality environment is even greater than for providing a map.

2.2 Communication for Robot Supervision

Monitoring, supervision, and recovery are capabilities that are related to both control and Human-Robot Interaction. Whether we consider assembly robots or tool machines, the existence of such capabilities is mandatory for operation in the real world and within possibly complex manufacturing chains. However, it is impossible to foresee all situations that may be encountered a priori to determine appropriate recovery strategies. Therefore, strong demands for learning such capabilities exist.

In contrast to tool machines, robots (such as assembly robots) are expected to be capable of both diagnostics and error recovery. In general, three classes of errors can be distinguished (Lopes and Camarinha-Matos, 1995b): system faults, external exceptions and execution failures. Execution failures are, for example, collisions, obstruction, part slippage from the gripper, part missing at some expected location, etc. External exceptions are abnormal occurrences in the cell environment that may cause execution failures. Examples are misplaced parts, defective parts, and unexpected objects obstructing robot operations. System faults are abnormal occurrences in hardware, software, and communication media. In summary, an execution supervision system must provide support for *dispatching, execution monitoring, failure diagnosis,* and *failure recovery.*

The main problem in this context is the acquisition of knowledge about the tasks and the environment. Even for a human domain expert it is difficult to specify appropriate mappings from sensor patterns to monitoring conditions, failure classifications, failure explanations, and recovery strategies. Providing explicit examples of erroneous behaviour is even more difficult and is especially sensitive to errors in the user's model of the robot and the task (Lopes and Camarinha-Matos, 1995a).

Supervision as described above, tries to find faulty operations, and reports these errors to the user. Instead of searching explicitly for errors, the performed actions and perceptions can be classified and matched with the desired goals. In case of mismatch, the user can be notified that some error occured. In addition, at each time the robot can tell the user in terms of perceived concepts what it is currently doing, i.e. it can communicate the result of applying an "operator".

2.3 Communication for Robot Programming

Usually, the design of robot programs involves a mixture of quantitative and qualitative modelling techniques as well as training procedures based on interaction with a human user (Fig. 2). Especially for users who are not experts in robot programming, **Programming by Demonstration (PbD)** (Cypher, 1993) has a considerable potential to become a suitable programming technique and to replace conventional robot programming languages (see, for example, (Groover, 1986) for an overview). **PbD** relies on demonstrations of the considered task. The demonstrations are used as the primary input and are the basis for a **learning process.**

PbD has been applied successfully in domains such as graphic editors (Lieberman, 1993), instructible software agents (Maulsby, 1994) and intelligent interfaces (Minton, 1995). **Robot Programming by Demonstration (RPD**, (Heise, 1989)) has been realized through a number of applications on different levels of both robot control and perception.

- Demonstrations were proven to be suitable for the acquisition of *new program schemata on task level* (Segre, 1989). In (Kuniyoshi et al., 1994), sequences of video images were analyzed in order to generate assembly plans. (Andreae, 1984) presented NODDY, a system which generates generalized programs by fusing several demonstrations. Single demonstrations and user intentions are the basis for the robot programs generated by the system described in (Friedrich et al., 1996).

- On the *control level*, demonstrations can be used as the basis for learning both, *open-loop and closed-loop elementary skills.* The acquisition of open-loop skills is mostly focused on the reconstruction of trajectories from a sequence of demonstrated states (positions) (Delson and West, 1994), (Ude, 1993).
  Systems supporting the acquisition of closed-loop elementary skills comprise acquisition techniques for manipulation tasks such as deburring (Asada and Liu, 1991) and assembly (Kaiser and Dillmann, 1996) as well as for vehicle control (Pomerleau, 1991) and autonomous robot navigation (Reignier et al., 1995), (Kaiser et al., 1996).

- Learning new perceptive skills for object and landmark recognition can also take place on several system control levels. (Accame and Natale, 1995) present an approach to learn sensor parameterizations from demonstrations. Learning active perception skills, i.e., the combination of actions and sensing for the purposes of object recognition, is the topic of work presented in (Klingspor et al., 1996).

In Machine Learning, **behavioural cloning** has become popular as a synonym for "skill acquisition via human demonstration". Typical applications are the cart-pole balancing tasks (Guez and Selinsky, 1988), (Dzeroski et al., 1995), as well as the work on "Learning to fly" (Sammut et al., 1992) and recent work on crane control (Urbancic and Bratko, 1994), (Bratko et al., 1995). In contrast to work in robotics, these approaches focus on the evaluation of a specific learning technique for cloning. Also **imitation learning** (Hayes and Demiris, 1994), (Demiris and Hayes, 1996) must considered in the context of Robot Programming by Demonstration. It is concerned with learning by imitating another agent (another robot), and as thus can be considered a special case of PbD. An important difference is that the actions that the robot performs due to the imitation mechanism form the basis for learning (Section 4.1). Therefore, the robot does not have to understand what the teacher is perceiving, and the communication with the teacher is minimal: the robot is learning by imitating the teacher and associating *its own actions* with *its own perceptions*. In order to achieve the imitation, additional mechanisms that translate observations into an appropriate internal model are needed. Based on this model of, e.g., the observed actions of the teacher, the robot can plan and execute appropriate actions. Since the teacher might be unaware of the robot pupil, the latter must be equipped with more sophisticated sensor systems (for example vision systems) than in PbD. However, the requirements on the actual interaction interface are less demanding.

3 Learning for Human-Robot-Communication

In the previous section, we presented three different aspects of Human-Robot-Communication, namely communication for task specification, monitoring, and learning. The approach described in this section enables the user on the one hand to give commands to the robot, for example task specifications, and on the other hand to get feedback from the robot about its activities. In contrast to map-based navigation, communication is not based on a fixed set of landmarks. It is, however, similar to human navigation in unknown environments based on events which can be reached during the act. An example of a sequence of commands is: "move through the door, rotate beside the door, move along the wall, and stop at the cupboard." Because the aim is not to develop a sophisticated natural language interface, the communication entities are PROLOG predicates. The statement above then can be translated into a sequence of PROLOG facts linked by their arguments (denoting time points):

```
move_through_door(T1, T2) &
rotate_in_front_of_door(T2, T3) &
```

move_along_wall(T3, T4) &
approach_cupboard(T4, T5).

Specifying the tasks that the robot has to perform is only one part of the communication. The same concepts that are used by the user to define tasks can be used by the robot to report about its activities. Therefore, the robot can use different levels of abstraction of our representation hierarchy (described in (Klingspor et al., 1996)) to enable the user to find reasons for the failure of a mission. E.g.:

"mission failed because
    move_through_door failed, because
        through_door failed.
    Instead, move_along_door was perceived."

where through_door is the perception of the robot when crossing a door, and move_along_door is the whole concept (including action and perception) of passing a door.

In contrast to other monitoring approaches, we do not try to find and classify the faulty situation in terms of error situations, but in terms of what the robot *has perceived*. Thus, the user has access to the state of the robot's mission at all times.

In cases of divergencies between expected and actually perceived observations, the system replans in order to find another way to reach the goal starting from the current state and perception.

3.1 Why is learning essential?

Depending on the knowledge of the user and the knowledge of the robot, interfaces between the user and the robot may be very different. If the robot is to be operated by an inexperienced user, a high-level-interface is necessary, where the entities of the language are tailored to the user and hence easily understandable by him. Our work only concerns verbal communication, based on PROLOG facts.

User and robot can only understand each other if the symbols used for communication describe the same concepts. The extension of the concepts, i.e. the objects that are covered by the concepts, must be the same. This can be easily achieved with map-based approaches. The user can point to objects and classify them, for example "this is a table, ...". Then an instance of an object can easily be found by moving to the corresponding position. In unknown environments, however, the robot must be capable of finding instances of concepts based on its sensors only, i.e. it must be able to classify its perceptions. In artificial systems, like medical expert systems, classification is based mainly on a few artificially discrete features. Discretization is often straightforward due to the large amounts of accumulated knowledge (past experiences) in that field. In contrast, here the input for classification is, at least on the lowest level of abstraction, a stream of continuous and noisy real valued measurements. In robot systems, the system developer usually analyses the data with different methods, combines them, generates new features, and searches for classification methods based on all these features (Rauber et al., 1993). This problem of "manual" symbol grounding is a very time consuming and hard task (Harnad, 1990). Because of their different sensor systems, robot and developer have different views of the world. For example, in our case the robot only has distance sensors (sonars) in contrast to the powerful human vision system which is capable of identifying very different complex objects. Features easily perceived by humans may be hard to be perceived by the robot. For example, a doorway can be described by a human using different convex corners that are building the door posts. Convex corners, however, are very difficult to detect with a sonar sensor, because the sensors get the echo only from one of the two walls of the corner. The other wall does not reflect the sonar beam due to the unfavourable angle between the wall and the beam. If the developer of the robot defines the concept of door crossing based on the corners building the door posts, the robot will rarely classify its perceptions correctly. It takes a lot of knowledge about the characteristics of the sensors and a lot of time to define good concept descriptions, even for experts. The problem arises, because the developer tries to transfer its own intensional description of a concept to the robot. It is, however, not necessary that the human and the robot share their *intensional* description, they only must have the same *extension* of the concept. Therefore, the robot should generate its own description of concepts, that might be very different from those descriptions expected by the user. That means, the robot must be capable to *learn*. If, for example, in certain situations the same erroneous specular reflection occurs every time, then this reflection can be used as a feature even if it is generally a weakness of the sensor. In this way the usefulness of the symbols for defining higher level concepts determines how more basic features are grounded. This way of grounding symbols is close to the way proposed by (Wrobel, 1991).

The necessity to use machine learning for sharing knowledge between human and robot has been mentioned by (Hiraki and Anzai, 1996), too. In their paper they describe a way to learn classifiers using feature abstraction, namely combining different features by basic arithmetical operations.

3.2 Combining actions and objects

In classical systems objects and actions are separated. Objects are seen as something to be manipulated and actions are operators defining the manipulation of objects. Humans, in contrast, represent events by relating objects to an action that is performed with these objects in a single concept. The couplings between actions and their corresponding objects were termed *predicates* by (Abelson, 1963), e.g., drinking from a cup, throwing a ball, moving through a doorway. Nelson states, that "..., young children must represent their own roles and the roles of other and be able to reciprocate actions of the other with actions of their own, ..." (Nelson, 1983). Objects should be represented as relations between actions and reactions, where an object implies a specific action, like a "ball implies throwing". Therefore, instead of representing just objects (or classes of objects – concepts) alone like "doorway", and representing actions alone like "move forward", operational concepts relate objects with typical operations, like "cross the door". In addition, "An object or category of objects may recur in a number of different events, in similar or different relationships" (Nelson, 1983). The category of doors may occur in different events like moving along them or moving through them, so we have different concepts relating different actions with different perceptions of the concept *door*.

This combination of object and associated action is implemented by integrating perceptual features and action features. In addition, this integration supports the classification of an object as an instance of a concept. In a conventional representation, a cup, for example, is defined by having a flat bottom and a handle and being concave. But it is easy to find arbitrarily many objects with these properties that are not cups because it is impossible to drink from them, e.g., if the handle bridges over the opening (De Jong and Mooney, 1986). Finding a complete description of a cup that excludes all exceptions is impossible because of the infinite number of these exceptions. This is known as the qualification problem (McCarthy and Hayes, 1969). So, how to define a cup suitably? The main issue of a cup is that you can drink from it. If a cup is defined as a receptacle from which drinking must be possible, the object classification can be verified by performing this action. If it is possible to drink from the perceived object, it can be classified as a cup. In this way, actions are integrated into concept descriptions and their recognition functions.

Giordana and Saitta have proposed to use *executable features* in concepts descriptions (Giordana and Saitta, 1990). These features are true for an object if a particular handling of this object is successful. For instance, the feature "movable" for a concept can be verified by moving that object. This is closely related to the arguments of (Pfeifer and Scheier, 1994), that perception is a process that needs action. Perceptual features require the integration of the action that is performed while the object is being perceived. If, for example, you are looking at a cup from above you cannot determine whether the bottom is flat or not. The perception is restricted by the action during which an object is perceived. Action features, in turn, require the integration of perception. Particular sensor patterns express the applicability conditions, i.e. the conditions for successful performance of the action, and the conditions for ending the action. In this way, an action is expressed by a set of perceptions.

3.3 Example: Integrating Human Robot Communication and Learning in a reactive planning system

In our scenario, learning and applying the learned knowledge are two different phases. Figure 3 illustrates these phases.

During the learning phase, the user presents some examples of concepts to the system by guiding the robot through the environment using a joystick. She then communicates to the system what she would like to call the situations she has just shown to the robot. These examples are the primary source of knowledge and supervised learning is applied at all levels of our representation hierarchy (Klingspor et al., 1996).

During the application phase, the learned knowledge is used to control the robot. The user specifies a task to be performed. Then, the plan-scheduler activates several planners with different strategies in parallel in order to find a suitable plan as fast as possible. This plan is represented in terms of operational concepts. The plan refinement and execution unit operationalize the abstract plan to executable features and abstract perceptual features. The executables features are sent to the robot to trigger some actions resulting in a set of sensor measurements. These are sent to the object identification unit, which infers features in parallel at all levels of the hierarchy. These features are sent to the plan-scheduler which updates the plan, and to the plan refinement unit, which tries to match the perception with the current plan. In this architecture the plan-scheduler is the interface between user and robot, providing reports about desired plans and performed operations and receiving new commands from the user.

3.4 Learning operational concepts

The first step when starting with learning from robot data is to decide which representation *formalism* will be used. It is important that the formalism is capable of representing temporal relations, relations between different sensors and sensor classes, and relations between perceptions and actions. In addition, the more knowledge about learnability within this formalism is available, the faster the problem of finding a learnable representation language and a learning algorithm can be solved. We have chosen restricted first order logic as our representation formalism. In contrast to attribute/value-formalisms it is capable of handling relations. Furthermore, methods and theory of *inductive logic programming* are available for this formalism.

The problem arising next is the definition of the representation *language*, i.e. the predicate symbols and their arguments. This, however, is a process that is influenced by the experience of using this language for learning. Hence, the whole modeling process is a cyclic process of

1. presenting real world examples to the robot,
2. calculating basic features,
3. generating features at the different levels of the representation hierarchy,
4. learning of concept descriptions at the different levels,
5. and evaluating learning and representation.

After each step of this process, especially after step 4 and 5, the result may yield changes in previous steps. In the following, we describe the decisions, that can be made at each of these steps. The basic techniques used in the different steps, as well as the concrete representation hierarchy, are described in detail in (Klingspor et al., 1996).

Presenting real world examples In this step, the desired concept is performed with a real robot in the real world. Different ways to control the robot exist: point-to-point-navigation, joystick-control, but also control by more complex media, or multi-modal. The approach of learning from communication, described in Section 4, emphases this step. Here, we abstain from a precalculation step. Therefore, we need very smooth data that can only be obtained from point-to-point navigation. A user demonstration performed with a joystick produces a path that is too rough.

Calculation of basic features The first step when abstracting from real valued numbers is finding sequences of measurements of a single sensor, where the change of sensed distance is approximately linear. To some extent, different parameters define abstract basic features. Which of the available parameter sets is best suited for learning can only be decided by evaluating the learning results.

Generating features for learning To be able to learn shorter and simpler rules, we use a hierarchical structure to represent concepts. We add two further levels to the level of perceptual features (describing perceptions of the whole robot): perceptions of single sensors and perceptions of sensor groups. Each feature can be described based on the features of the next lower levels. The features at the level of single sensors are based on basic features.

The next step is to decide what exactly will be represented at each level. In (Klingspor et al., 1996), *we* define four features: concave and convex corners, jumps, and lines. We already mentioned in Section 3.1 why this decision is dangerous, and experiments showed that these features cannot be reliably perceived by the ultrasound sensors that we used.

The opposite approach is to let the system find out which sequences of basic features appear most often. These sequences are given new names and they are used as features for classifying higher level concepts. Whereas the former learning task was supervised learning this is an example of unsupervised learning. Experiments have shown that these sequences are easy to find. However, they cannot be used to discriminate higher level concepts, since they are generated without regard for what they should be used for.

Therefore, we now apply an intermediate approach. For each feature at the perceptual feature level (e.g. through_door), we define a separate own sensor and sensor group feature, s_through_door and sg_through_door. Then, the learning task is to find common sequences of basic features within the time interval of different examples of the goal concept. The results presented later in this section show, that this approach is more successful than the one described in (Klingspor et al., 1996).

Learning concept descriptions The next step is learning itself. At each level of the hierarchy concept descriptions for each concept are learned. In principle, each ILP-method can be used for this task. However, each method deals with special characteristics of the domain, and the most important property of our learning task is the huge amount of noisy data. Therefore, some learning methods are of special interest.

Goebel presented a very specialized algorithm for learning sequences from positive examples (Goebel, 1996). Because this method reduces the data syntactically and applies a very efficient method for generating and evaluating hypotheses, this algorithm is up to ten times faster than the algorithms applied before. Efficiency is traded off against flexibility of the algorithm, only a single structure of rules can be learned, and since only positive examples are taken into account.

Without regarding negative examples, it is hard to decide when a hypothesis is specific enough to discriminate instances of a concept from non-instances but general enough to cover many instances, because this measure can only depend on the coverage, but not on the correctness of a hypothesis. Therefore, we additionally performed learning from positive and negative examples with GRDT (Klingspor et al., 1996). We applied the *closed world assumption* as a specific method of this type of learning. GRDT tries to learn rules covering exactly the given instances of the concept, but deriving no further instances in other traces. Evaluating hypothesis with respect to the closed world assumption is much more time consuming, because the inferential closure over the hypothesis must be build and intersected with the positive examples. In addition, GRDT is more flexible than Goebel's method, because the structure of the learnable rules can be defined by the user and is not predefined by the method.

Evaluation The last step of an interaction through the modelling process is the evaluation of the learning result. We applied coverage and correctness as criteria, where coverage here denotes the number of covered positive examples divided by the number of given examples, and correctness is the number of covered positive examples divided by the total number of derived facts. In general, there exists a trade off between these criteria. More general learning results yield in a higher coverage and more special learning results yield in a higher correctness. The higher both criteria are, the better is the whole modeling process. As long as the result is insufficient, some of the decisions made during the modeling process have to be revised, and the modelling loop restarts.

To support the decision we made, we present the following experiments. Input for learning is the data described in (Klingspor et al., 1996). Lines with MLJ as indicator describes results with the predefined set of sensor features, these are the results of (Klingspor et al., 1996). MG describes experiments with the new method of Michael Goebel for learning from positive examples with the new representation. GRDT describes the results when additionally applying the closed world assumption using GRDT.

Table 1 shows the coverage and accuracy of the different learning results. The results strongly support our new representation language coverage and correctness of both learner using the new representation are higher than in the MLJ-experiments. Furthermore, the choice of learning methods capable to handle negative examples or the closed world assumption again yields in better results.

3.5 Discussion

In this section, we presented an approach to automatically acquire knowledge for task specification and execution supervision. The learned knowledge provides the basic symbols for communication between user and robot. In both task specification and supervision, communication via these symbols differs from other approaches.

Task specification in unknown environments is achieved by using objects which are previously unknown to the robot, but which can be classified as instances of a learned operational concept. In most systems, the tasks are given in terms of landmarks with known positions, or in unknown environments, by accompanying the system during execution. This way, we can give abstract instructions a priori, and the system itself plans which additional operators are necessary to execute the given instruction. For example, if the user commands the robot to leave the room, the robot has to find the door first (by moving to a wall and follow the wall until it passed a door opening), then to position itself exactly in front of the door (because PRIAMOS, the robot used in our experiments, is large), and then to cross the door. Clearly, the planned sequence of operators can be interrupted by the user at any time, for example if the robot enters insecure areas or runs into loop. But typically the robot performs the given task autonomously.

To monitor the robot's activities, we introduce a new approach, too. Most monitoring approaches regard a sequence of operations as correctly performed, as long as no error can be classified. These systems cannot classify perceptions and actions as correct instances of the given operation. In contrast to that, in our system each action is monitored. All perceptions and actions are reported in abstract terms, and in human understandable form. This way, each difference between planned operation and observed operation (both represented by operational concepts), can be seen as an error. The observation can not only be used as new starting point for planning, but also to tell the user what the robot thinks it has done. This is very different from just telling the user which operator is currently being applied.

An important point of our approach is that the concepts used for communication (which are used by human and robot in common) are derived using machine learning techniques. This not only reduces the modeling effort, but it also increases the prospects that the intermediate concepts used in our representation hierarchy correspond better to the sensors of the robot.

In addition, learning allows an end user who has not received any special training, to teach the robot to enlarge its capabilities by adding new concepts. Such a teaching, however, requires interaction between user and robot not only during performance, but also for providing examples to learn from: the user *shows* the robot which actions and which perceptions are encapsulated in a certain concept by performing the corresponding operation with the robot. How communication between robot and user can be used for learning will be shown in the following section.

4 Learning from non-verbal Communication

Until now we have argued why learning is necessary to enable the robot to translate user commands into low level robot operations, and vice versa, to classify the robot's perceptions and actions for feedback. The main purpose within this context is to make the robot smarter in order to reduce the load on the user, both during robot deployment (i.e., while preparing the environment for the robot) and robot monitoring and supervision. However, to let the robot develop useful skills in communication, perception, and action, it must be provided with examples of appropriately skillful behaviour. Where do these examples come from?

4.1 Bottom-Up Skill Acquisition and Imitation Learning

If skills are acquired in a **bottom-up** manner, the robot starts from explicit instruction, i.e., from **examples** of skilled behaviour. Actual skills are developed by means of **learning** from these instructions, e.g., via generalization over the situations that are covered by the instructions*.

For humans, the acquisition of skills without the availability of explicit knowledge has been reported in (Stanley et al., 1989), (Lane, 1988). This is the psychological basis for PbD, Skill Acquisition, and behavioural cloning.

Imitation in humans and animals involves cognitive operations that are complex and not well understood (Heyes, 1993). Despite the apparent complexity, it is useful for robots to have such capacity since learning by imitation offers certain desirable characteristics, most notably the ability to learn autonomously by observing and imitating without interrupting the expert while she is performing the task to be learned. Imitation greatly reduces demands from both the user and the interaction interface. It achieves that by introducing an additional level of complexity in the robot's control system: the mechanism for matching observed user movements with equivalent robot movements. (Demiris and Hayes, 1996) have proposed an imitation mechanism inspired by experimental results of research in early infant development (Meltzoff and Moore, 1989). This mechanism views the imitation process as a dynamic system: when the robot decides or is requested to imitate, it calculates simple geometric relationships between parts of its body and the user's body (such as distances and angle differences between the postures of the two agents) and attempts to move its body parts in a way that maintains these relationships, while the user's movements are essentially altering them. For more details about the imitation mechanism and its relation with mechanisms hypothesised to underlie human imitation see (Demiris and Hayes, 1996).

There are two sources of data that are used in the learning process under this paradigm. The environmental state as perceived by the robot forms the

*Recent research in cognitive science (Van Lehn, 1996) indicates that generalization may not be an automatic process but needs to be triggered or guided by a teacher. However, in the case of inductive Machine Learning, generalization is the ultimate goal that's being pursued. Especially in bottom-up skill learning, generalization is mandatory.

first source of data. Motor actions that are performed due to the imitation of a demonstrator form the second source of data. Imitation learning has been used primarily in order to learn pairings of the form environmental_state - action(s)_to_be_followed (situation-action rules) (Hayes and Demiris, 1994). It can also be used to learn the meaning of certain motor-related words by associating the words transmitted with the actions performed (or the resulting end states) (Billard, 1996). Imitation is also an appropriate method for learning additional motor skills.

In the type of tasks that imitation has been used till now, the principal learning technique has been single-trial associative learning. Any type of associative learner can be used (pattern associator, Willshaw Nets etc). After the required knowledge has been learned and stored as production rules of the form *if situation then action*, in the application phase the robot is constantly attempting to match the environmental conditions that it encounters with the stored patterns learned during the training phase. If a suitable match is found, the corresponding actions are executed.

Robot imitative mechanisms are relatively new and learning by imitation has only been demonstrated in the context of Robot-Robot Interaction and not yet in the context of Human-Robot Interaction. In the robot-robot context, the experiments reported in (Hayes and Demiris, 1994) have demonstrated how a robot can learn to negotiate the different corners of a maze by imitating the movements of a teacher robot. A teacher robot knowledgeable of the appropriate actions to perform is placed in a maze, and a learner robot is placed behind it and asked to imitate the teacher. While the teacher robot navigates through the maze, the learner robot follows it and it imitates its movements. However, while it is doing that, it associates the environmental state perceived through its sensors (i.e. the wall configuration) with the actions it is performing due to its imitation of the teacher robot (different types of turning actions), effectively learning what is the right action to do under certain environmental situations.

To achieve the robot imitation abilities that are required for Human-Robot Communication, (Demiris and Hayes, 1996) have proposed the more general imitation mechanism which views the problem as a dynamic process where the learner participates in an active role since actions are viewed as relationship changes. This has the additional significance that this process can become *bidirectional* with the robot imitator being able to detect (as humans are able to) when *it is being imitated*: relationship changes caused by *its own actions* are being restored *by the other agent*. This sensitivity to being imitated arises early in the life of human infants and is believed to underlie the development of early communication abilities in infants (Nadel and Fontaine, 1989).

4.2 Example: Skill Acquisition from Human Demonstration

Two important findings result from the analysis of human skill acquisition and learning by imitation. First, skill acquisition is a **process** that comprises, at the very least, the phases of initial knowledge acquisition (e.g., via examples), initial learning, application and adaptation. Second, skill learning from examples without having explicit knowledge about the task, is a concept that can be found in human skill acquisition, so it seems to be a valid approach to learn robot skills from examples, too.

4.2.1 The learning task

For a given state $\boldsymbol{x}(t)$, a skilled system (the robot) should perform a goal-oriented action $\boldsymbol{u}(t)$. The action performed should be the result of a competent decision, i.e., it should be optimal with respect to an evaluation criterion (a reward) $r(\boldsymbol{x}(t))$ that is related to the goal to be achieved. Essentially, a skill s is therefore given through a control function

$$C_s : \boldsymbol{u}(t) = C_s(\boldsymbol{x}(t))$$

that implicitly encodes the goal associated to the skill and produces in each state $\boldsymbol{x}(t)$ a competent action $\boldsymbol{u}(t)$, and a reward function $r_s(\boldsymbol{x}(t)) \mapsto [r_{min}, r_{max}]$ that evaluates the state $\boldsymbol{x}(t)$ w.r.t. the goal. To allow for execution monitoring, an error criterion $e_s(\boldsymbol{x}(t))$ is also required.

If the skill application involves to move the system along a trajectory $(\boldsymbol{x}^*(0), \ldots, \boldsymbol{x}^*(T))$ in the state space, a termination criterion $t_s(\boldsymbol{x}(t)) \mapsto \{0, 1\}$ must also be present (see Fig. 4). The **learning task** is therefore to build all functions C_s, r_s, e_s, t_s from the user-given examples.

In the case of elementary action skills and elementary sensing skills, the state $\boldsymbol{x}$ is represented as a sequence of perceptions $\boldsymbol{y}$, i.e., $\boldsymbol{x}(t) = (\boldsymbol{y}(t - d), \ldots, \boldsymbol{y}(t - d - p)), d, p \geq 0$, while the action $\boldsymbol{u}(t)$ is directly executable by the robot (e.g., as a primitive motion or a one-shot parameterization of a sensor) without requiring any further processing on a level above or equal to the skill level. The skill is therefore **operational** with respect to the robot. The error criterion is given via boundary conditions on the state variables (e.g., as a hyperintervall containing sensor measurements that are assumed to be safe). The termination criterion is defined in a similar manner for the goal state.

4.2.2 The Process Model of Skill Acquisition

Since one of the main characteristics of examples provided via human demonstration is the varying quality of these examples (Kaiser et al., 1995), both example quality assessment (to provide the user with feedback) and on-line optimization of the initially acquired skill become important issues.

Thus, the skill acquisition process (Fig. 5) requires more than just example generation, "strategy extraction" (learning), and skill application. In the course of skill acquisition, support of the human teacher may be mandatory only in some phases:

1. The teacher identifies the need to acquire a new skill.
2. The teacher selects the scenario for the generation of the examples and performs a demonstration of an elementary skill. Alternatively, a strategy for autonomous experimentation including boundary conditions on perceptions and actions, and an evaluation function can be specified.
3. The teacher adds a symbolic interpretation to the learned skill or links it explicitly to a description of the context (e.g., via a map of the environment) in which the skill shall be executed.
4. The teacher provides a consistent on-line evaluation of the robot performance which consists at least of a *good/no good* feedback at the end of the application of the newly acquired skill.

Then, the human teacher is still heavily involved in the robot's learning process. However, the cumbersome task of explicit programming has been replaced by *verbal communication* on a high level of abstraction and *non-verbal communication* by means of demonstrations. To support this kind of high-level interaction, we also require that the parameterization of any of the algorithms for skill acquisition may be done automatically (Kaiser and Dillmann, 1996). We must make sure that the user doesn't have to provide any detailed information about the learning process. If, however, the user wants to give additional input, this should – of course – be used.

4.2.3 Acquisition of a force-control skill

As an example of a skill we use the "force control" skill, i.e., the ability to apply a constant force to a work piece. Examples of other skills, both related to manipulation and to navigation, can be found in (Kaiser and Dillmann, 1996), (Kaiser et al., 1996). In this case, the task for the human user was to bring the tip of the robot's gripper into contact with the surface of a workpiece and to apply a constant force F_z of $10[N]$.

The samples recorded during the demonstration (see Fig. 6) are preprocessed, relevant sensors and degrees of freedom are identified, and RBF networks for representing C_s and r_s are generated (for details see (Kaiser and Dillmann, 1996)). Fig. 7 shows how the very same robot used to record the example, now controlled by the off-line generated neural network, is able to apply a constant force to the workpiece surface. However, the applied force (approx. $9[N]$) is different from the desired one ($10[N]$), due to the non-optimal demonstration. Therefore, adaption is necessary. Note also that the whole process of generating the initial controller for this skill so far has been performed automatically.

For adaptation, the robot was initially exposed to the same workpiece again, such that only the offset contained within the demonstration data had to be corrected (Fig. 8).

More significant improvements occur if the robot's environment is changed. Figures 9 and 10 show the performance of the initial and an adaptive controller, if the force is to be applied to a more rigid workpiece than during training. While the initial controller keeps oscillating, the adaptive one is able to reduce the oscillations significantly, with the remaining level of oscillation being due to the deficiencies of the robot. The Puma 260 in use is controlled at about $30[Hz]$ in cumulative ALTER mode, resulting in a minimal positional offset of $0.1[mm]$.

4.3 Discussion

Learning from communication and especially learning from demonstration differs in several aspects from learning for communication. Most important, the robot's performance in terms of appropriateness of its physical actions depends on the quality of the teacher's demonstration. The robot learns a concrete and elementary skill from its human (or robotic) teacher. Should the teacher perform badly, the robot will perform badly, too. In learning for communication, the robot may not correctly understand its human user if it learns something stupid. However, it will still be capable to exhibit satisfactory concrete performance. Therefore, the following requirements must be met when learning from demonstration:

1. The human teacher must be **competent** with respect to the skill to be demonstrated, both in the PbD and the imitation case.
2. The system that should acquire the skill must provide adequate sensing and acting capabilities (sensors, degrees of freedom).

Then, it is possible to learn usable (although not optimal) skills even from a single demonstration. These initial skill can be further adapted. Depending on the amount of information available for adaptation, we may obtain the following results:

1. If only the minimally necessary information (a binary evaluation at the end of the skill application) is available, adaptation may result in making the skill more widely and more safely applicable. It will, however, not result in an "optimal" skill.

2. If we are able to provide a more detailed evaluation function or even a target trajectory in the state space, skill adaptation may result in skills that are optimal (w.r.t. this function) but have required only very little effort to design.

If things go wrong and the human user proofs to be not competent*, learning from human demonstrations will not work satisfactorily. In particular, we may experience the following problems:

1. If the human user is not capable to demonstrate the skill correctly, trying to learn something from these demonstrations will require to construct huge state descriptions, in order to account for inconsistencies in the data.

2. Even if we are then able to learn what might have been the human's strategy, this doesn't help, since this strategy has been not successful.

3. Finally, even if the strategy was correct, and we are not able to let the system apply skills only in appropriate contexts or to define at least qualitatively correct evaluation functions, successful skill application and adaptation is not possible.

5 Conclusion

Throughout this paper, the relationship between learning and communication in Human-Robot Interaction has been discussed. The need for a vocabulary (an ontology) that is common to the user and the robot, the need to extend this vocabulary according to the needs of the user, and the desire for sophisticated programming techniques have been made explicit. In any of these cases, learning plays a fundamental role, since every increase in smartness on the side of the robot will reduce the burden on the user. While both approaches differ in their treatment and use of Human-Robot

*It should be noted that especially the cart-pole balancing task (Guez and Selinsky, 1988) is a task that proved to be extremely difficult for humans to demonstrate (in simulation). Even if they were able to balance the pole, they were never capable to stabilize the cart-pole system in the desired target state (cart and pole in zero position). Trying to learn something from these demonstrations resulted in huge state descriptions and poor performance, whereas learning from only approximately working rule-based controllers was straightforward and successful.

Communication, they can benefit from each other. In both cases, the system learns from communication between human and robot. The learning result can be used for communication. The approaches differ, however, in concentrating on only one of the two aspects. Therefore, further work is necessary to couple both approaches more closely, in order to enhance the respective weak parts of each individual approach.

We believe that the assumptions we have made, i.e., to rely on the ability of a human teacher to demonstrate a solution to a given task, to use natural language symbols to describe the solution in part or as a whole, and to provide at least a qualitatively correct evaluation of the robot's performance, are realistic. We cannot expect that the communication, action, and perception skills acquired via an interactive learning approach are comparable with those originating from an in-depth task analysis and explicit robot programming. However, especially if robots are to become consumer products, they will be exposed to users who are not at all familiar with computers or robots. For such users, explicitly programming their robot according to their personal requirements is not an option, whereas teaching by showing and annotating definitely is.

Acknowledgement

This work has been partially funded by the European Community under the ESPRIT III project B-Learn II (P7274) and the BRITE-Euram III project BE-1564 Skill-MART, and the Ministry for Sciences and Research of the German federal state Nordrhein-Westfalen.

We would like to thank all persons involved at the different institutes for their contributions, namely Katharina Morik, Anke Rieger, Stefan Sklorz, Stephanie Wessel, Michael Goebel, and Eckart Zitzler from Dortmund; Michael Deck, Holger Friedrich, Albert Retey, and Frank Wallner from Karlsruhe; and Gillian Hayes from Edinburgh. Many thanks to Thorsten Joachims for proofreading the paper.

References

Abelson, R. (1963). Hot cognition. *Computer Simulation of Personality.*

Accame, M. and Natale, F. D. (1995). Neural tuned edge extraction in visual sensing. In *Third European Workshop on Learning Robots*, Heraklion, Kreta.

Agre, P. and Chapman, D. (1990). What are plans for? In Maes, P., editor, *Designing Autonomous Agents*, pages 17–34. MIT Press.

Andreae, P. M. (1984). Constraint limited generalization: Aquiring procedures from examples. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 6 – 10.

Anzai, Y. (1993). Human-computer-interaction in multianget environments. In *Procs. of HCI International'93*, pages 2–7, New York. Elsevier.

Asada, H. and Liu, S. (1991). Transfer of human skills to neural net robot controllers. In *IEEE International Conference on Robotics and Automation*, pages 2442 – 2448, Sacramento.

Asoh, H., Motomura, Y., Hara, I., Akaho, S., Hayamizu, S., and Matsui, T. (1996). Aquiring a probalistic map with dialogue-based learning. In Hexmoor, H. and Meeden, L., editors, *ROBOLEARN '96: An Intern. Workshop on Learning for Autonomous Robots*, pages 11–18.

Billard, A. (1996). Learning to speak through imitation for social robots. Master's thesis, University of Edinburgh, Scotland, UK.

Birk, A. (1996). Learning to survive. In Klingspor, V., editor, *Procs. of the 5th European Workshop on Learning Robots*, pages 1–8, Bari, Italy.

Bratko, I., Urbancic, T., and Sammut, C. (1995). Behavioural cloning: Phenomena, results, and problems. In *5th IFAC Symposium on Automated Systems based on Human Skill*, Berlin.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The psychology of human-computer interaction.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Cypher, A. I. (1993). *Watch what I do – Programming by Demonstration.* MIT Press, Cambridge, Massachusetts.

De Jong, G. and Mooney, R. (1986). Explanation-based-learning: An alternative view. *Machine Learning*, 2(1):145–176.

Delson, N. and West, H. (1994). The use of human inconsistency in improving 3D robot trajectories. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1248 – 1255, München.

Demiris, J. and Hayes, G. (1996). Imitative learning in robots and humans: Mechanisms and architectures. In Klingspor, V., editor, *Fifth European Workshop on Learning Robots*, Bari. Italy.

Dillmann, R., Kaiser, M., Klingspor, V., Morik, K., and Wallner, F. (1995). Teaching and understanding intelligent service robots: A Machine Learning Approach. In *19. Deutsche Jahrestagung für Künstliche Intelligenz (KI '95 Workshop: Informationsverarbeitung in Servicerobotern)*, Bielefeld, Germany. Springer Verlag.

Dzeroski, S., Todorovski, L., and Urbancic, T. (1995). Handling real numbers in ILP: A step towards better behavioural clones. In *European Conference on Machine Learning*, pages 283 – 286, Heraklion, Kreta.

Förster, S., Lobing, H., and Peters, K. (1995). Architekturkonzeption eines teilautonomen Montageroboters. Report – Situierte Künstliche Kommunikatoren 95/10, Universität Bielefeld, Sonderforschungsbereich 360.

Friedrich, H., Münch, S., Dillmann, R., Bocionek, S., and Sassin, M. (1996). Robot programming by demonstration: Supporting the induction by human interaction. *Machine Learning*, 23:163 – 189.

Giordana, A. and Saitta, L. (1990). Abstraction – a general framework for learning. In *Procs. AAAI – Workshop on Automatic Generation of Approximations and Abstractions*, pages 245–256.

Goebel, M. (1996). Lernen intervallbezogener Merkmale eines mobilen Roboters. Master's thesis, Universität Dortmund.

Groover, M. (1986). *Industrial Robotics - Technology, Programming and Applications*. McGraw-Hill.

Guez, A. and Selinsky, J. (1988). A neuromorphic controller with a human teacher. In *IEEE International Conference on Neural Networks*, pages 595 – 602, San Diego.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

Hayes, G. and Demiris, J. (1994). A robot controller using learning by imitation. In *International Symposium on Intelligent Robotic Systems*, pages 198 – 204, Grenoble.

Heise, R. (1989). Demonstration instead of programming: Focussing attention in robot task acquisition. Research report 89/360/22, Department of Computer Science, University of Calgary.

Heyes, C. (1993). Imitation, culture and cognition. *Animal Behaviour*, 46:999–1010.

Hiraki, K. and Anzai, Y. (1996). Sharing knowledge with robots. *Int. Journal of Human-Computer Interaction*, 8(3):325–342.

Kaiser, M., Deck, M., and Dillmann, R. (1996). Acquisition of basic mobility skills from human demonstrations. *Studies in Informatics and Control*.

Kaiser, M. and Dillmann, R. (1996). Building elementary robot skills from human demonstration. In *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA.

Kaiser, M., Friedrich, H., and Dillmann, R. (1995). Obtaining good performance from a bad teacher. In *International Conference on Machine Learning, Workshop on Programming by Demonstration*, Tahoe City, California.

Klingspor, V., Morik, K., and Rieger, A. (1996). Learning concepts from sensor data of a mobile robot. *Machine Learning Journal*, 23(2/3):305–332.

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822.

Lane, N. E. (1988). *Skill Acquisition rates and patterns – issues and training implications*. Recent research in Psychology. Springer–Verlag, Berlin, Heidelberg, New York.

Längle, T., Lueth, T., Herzog, G., Stopp, E., and Kamstrup, G. (1995). KANTRA – a natural language interface for intelligent robots. In Rembold, U. and Dillmann, R., editors, *Intelligent Autonomous Systems IAS-4*, pages 365–372. IOS Press.

Lieberman, H. (1993). MONDRIAN: A teachable graphical editor. In Cypher, A. I., editor, *Watch what I do*. MIT Press, Cambridge, Massachusetts.

Lopes, L. S. and Camarinha-Matos, L. (1995a). Planning, training and learning in supervision of flexible assembly systems. In *Procs. of Conf. on Balanced Automation Systems*.

Lopes, L. S. and Camarinha-Matos, L. M. (1995b). Inductive generation of diagnostic knowledge for autonomous assembly. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan.

Maulsby, D. (1994). *Instructible agents*. PhD thesis, University of Calgary, Canada.

McCarthy, J. and Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 5:463–502.

Meltzoff, A. and Moore, M. (1989). Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms. *Developmental Psychology*, 25(6):954–962.

Minton, S. (1995). Quantitative results concerning the utility of explanation–based learning. In Ram, A. and Leake, D. B., editors, *Goal-driven learning*, pages 55–82. A Bradford Book, The MIT Press, Cambridge Massachusetts, London England.

Nadel, J. and Fontaine, A. M. (1989). Communicating by imitation: a developmental and comparative approach to transitory social competence. In Schneider, B., Attilli, G., Nadel, J., and Weissberg, R., editors, *Social Competence in Developmental Perspective*. Kluwer Academic Publishers.

Nelson, K. (1983). The derivation of concepts and categories from event representations. In Scholnick, E. K., editor, *New Trends in Conceptual Representation: Challenges to Piaget's Theory?*, chapter 6, pages 129–149. Lawrence Erlbaum Associates, Hillsdale, NJ.

Pfeifer, R. and Scheier, C. (1994). From perception to action: The right direction? In Gaussier, P. and Nicoud, J.-D., editors, *From Peception to Action*, pages 1–11, Los Alamitos, CA. IEEE Computer Society Press.

Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 15:88 – 97.

Rauber, T. W., Barata, M. M., and Steiger-Garção, A. S. (1993). A toolbox for analysis and visualization of sensor data in supervision. In *Proc. of the Int. Conf. on Fault Diagnosis*, Toulouse, France.

Reignier, P., Hansen, V., and Crowley, J. (1995). Incremental supervised learning for mobile robot reactive control. In *Intelligent Autonomous Systems 4*, pages 287 – 294, Karlsruhe.

Roßmann, J. (1995). Virtual reality as a control and supervision tool for autonomous systems. In Rembold, U., Dillmann, R., Hertzberger, L., and Kanade, T., editors, *IAS-4, Proc. of the 4th Intern. Conference on Intelligent Autonomous Systems*, pages 344–351. IOS Press.

Saffiotti, A., Konolige, K., and Ruspini, E. (1995). A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1/2):481–526.

Sammut, C., Hurst, S., Kedzier, D., and Michie, D. (1992). Learning to fly. In *Procs. of the 9th Intern. Conf. on Machine Learning*, pages 385–393.

Schraft, R. D. (1994). *Serviceroboter - ein Beitrag zur Innovation im Dienstleistungswesen*. Fraunhofer -Institut für Produktionstechnik und Automatisierung (IPA).

Segre, A. M. (1989). *Machine Learning of Robot Assembly Plans*. Kluwer Academic Publishers.

Stanley, W., Mathews, R. C., Buss, R. R., and Kotler-Cope, S. (1989). Insight without awareness: on the interaction of verbalization, instruction and practice in a simulated process control task. *The Quarterly Journal of Experimental Psychology*, 41A(3):553 – 577.

Steels, L. (1994). The artificial life roots of artificial intelligence. *Artificial Life*, 1(1).

Stewart, J. (1995). The implications for understanding high-level cognition of a grounding in elementary adaptive systems. *Robotics and Autonomous Systems*, 16:107 – 116.

Ude, A. (1993). Trajectory generation from noisy positions of object features for teaching robot paths. *Robotics and Autonomous Systems*, 11:113 – 127.

Urbancic, T. and Bratko, I. (1994). Reconstructing human skill with machine learning. In *European Conference on Artificial Intelligence*, pages 498 – 502, Amsterdam.

Van Lehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, 47:513 – 539.

Wrobel, S. (1991). Towards a model of grounded concept formation. In *Proc. 12th International Joint Conference on Artificial Intelligence*, pages 712–719, Los Altos, CA. Morgan Kaufman.

Learning Task	Method	Coverage	Correctness
Sensor features			
	MLJ	28%	53%
	MG	73%	33%
	GRDT	73%	45%
Sensor group features			
	MLJ	85%	95%
	MG	92%	61%
	GRDT	87%	98%
Perceptual features			
	MLJ	70%	70%
	MG	49%	76%
	GRDT	97%	100%

Table 1: Experiments for learning operational concepts.

Figure 1: Bidirectional communication in Human-Robot Interaction.

Figure 2: Approaches to the Design of Robot Programs.

Figure 3: The abstract performance model.

Figure 4: The role of C_s, r_s, e_s and t_s during the application of an elementary action skill.

Figure 5: Different phases of skill acquisition, including skill application and refinement. Gray arrows indicate feedback loops. Only some phases are permitted to **require** user interaction.

Figure 6: Raw data sampled from a human demonstration of the CONTROL FORCE skill. The commanded motions of the robot are given in $1/100[mm]$, the forces are measured in $[N]$.

Figure 7: Performance of the robot controlled by an off-line trained network. F_z in $[N]$.

Figure 8: Skill adaptation throughout five runs. F_z in $[N]$.

Figure 9: Initial skill confronted with rigid workpiece. Force is shown in $[N]$.

Figure 10: Skill adaptation while being confronted with rigid workpiece. Force is shown in $[N]$.

Figure 1:

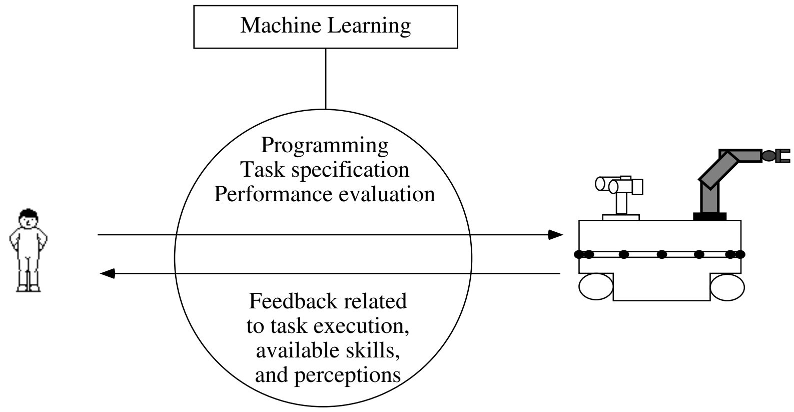

Machine Learning
Programming
Task specification
Performance evaluation
Feedback related
to task execution,
available skills,
and perceptions

Figure 2:

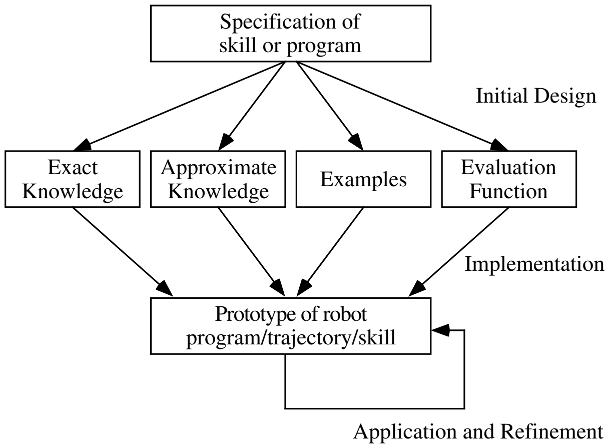
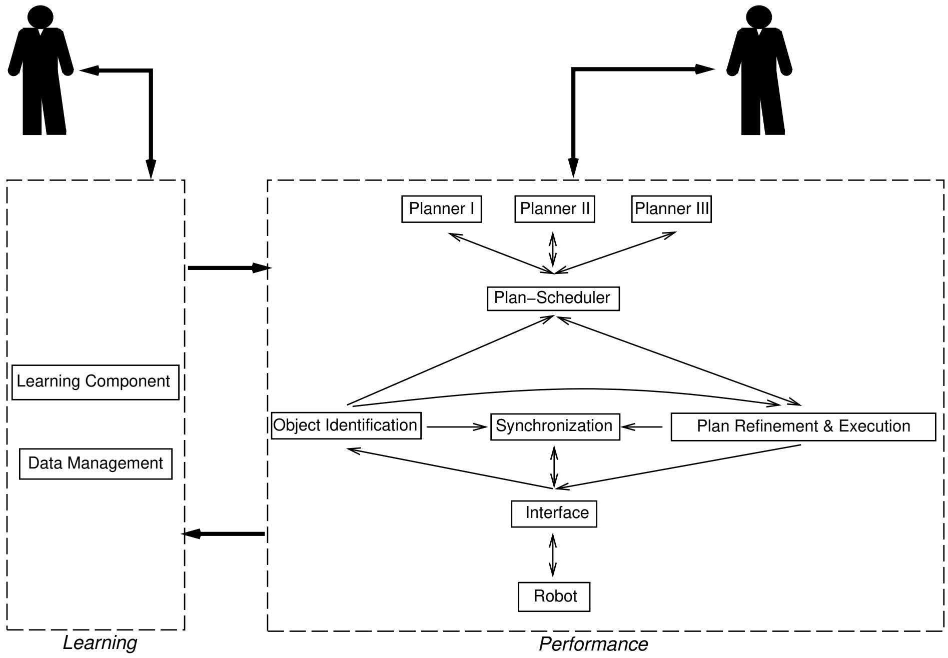
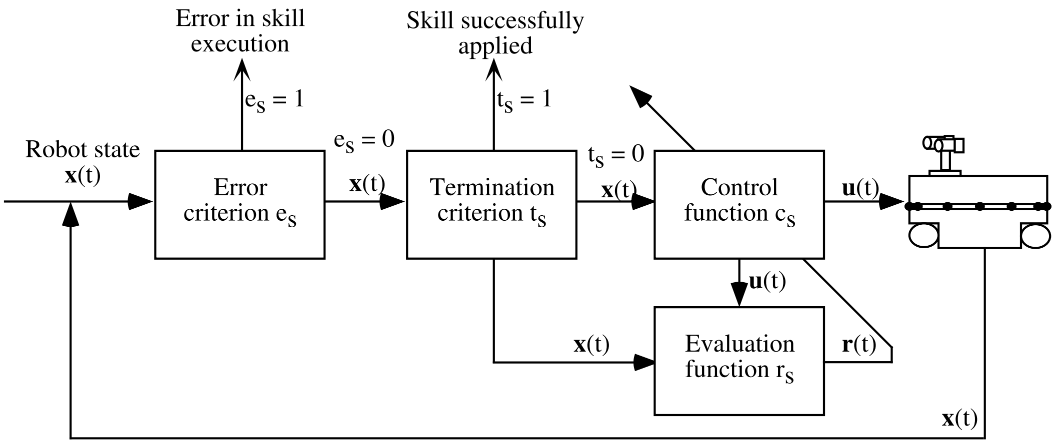

Specification of skill or program
Initial Design
Exact Knowledge
Approximate Knowledge
Examples
Evaluation Function
Implementation
Prototype of robot program/trajectory/skill
Application and Refinement

Figure 3:



Planner I
Planner II
Planner III
Plan−Scheduler
Learning Component
Data Management
Object Identification
Synchronization
Plan Refinement & Execution
Interface
Robot
Learning
Performance

Figure 4:



Error in skill
execution
Skill successfully
applied
$e_S = 1$
$t_S = 1$
$e_S = 0$
$t_S = 0$
Robot state
x(t)
Error
criterion e_S
x(t)
Termination
criterion t_S
x(t)
Control
function c_S
u(t)
u(t)
x(t)
Evaluation
function r_S
r(t)
x(t)

Figure 5:

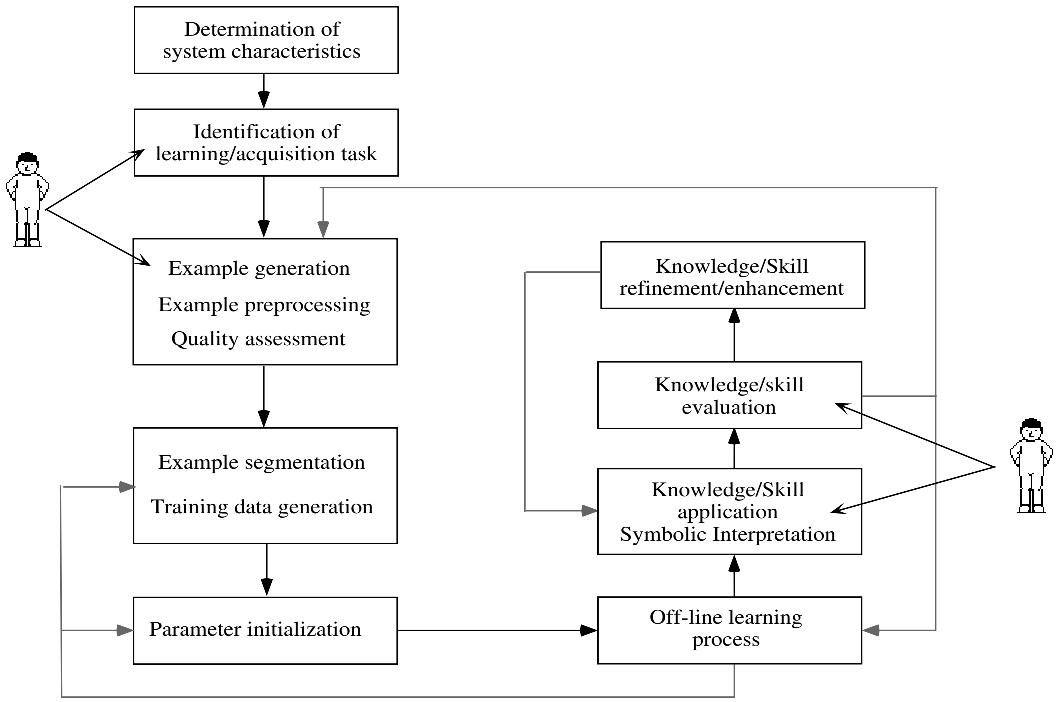
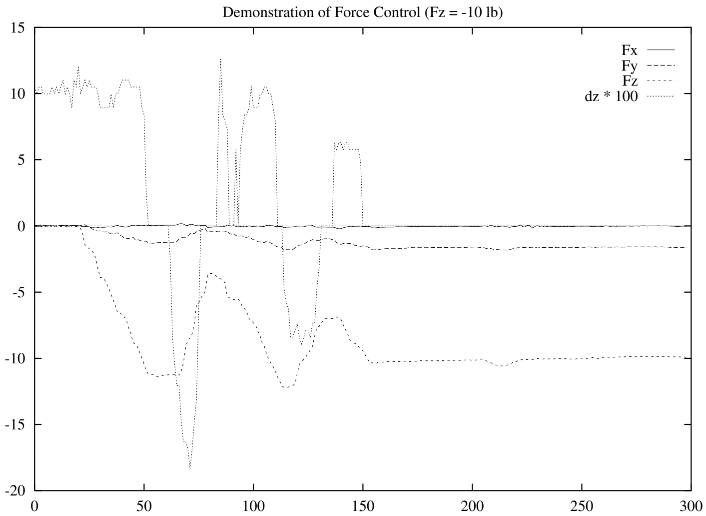
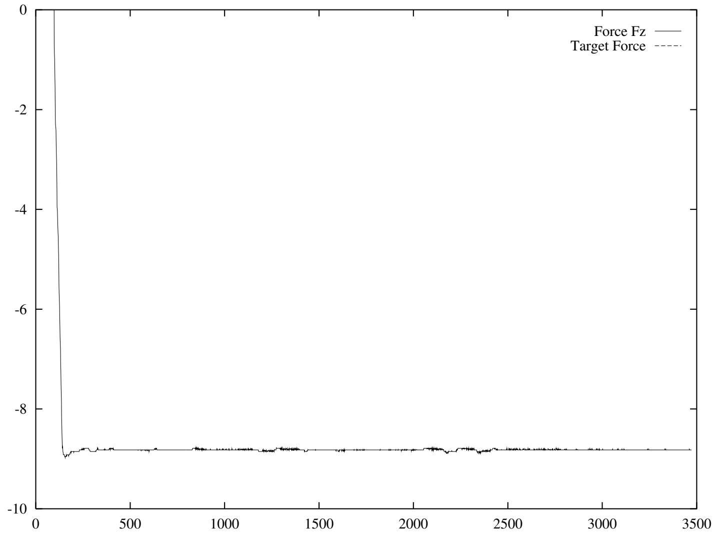

Determination of system characteristics
Identification of learning/acquisition task
Example generation
Example preprocessing
Quality assessment
Example segmentation
Training data generation
Parameter initialization
Knowledge/Skill refinement/enhancement
Knowledge/skill evaluation
Knowledge/Skill application
Symbolic Interpretation
Off-line learning process

Figure 6:



Demonstration of Force Control (Fz = -10 lb)
Fx
Fy
Fz
dz * 100
15
10
5
0
-5
-10
-15
-20
0
50
100
150
200
250
300

Figure 7:



Force Fz
Target Force
0
-2
-4
-6
-8
-10
0
500
1000
1500
2000
2500
3000
3500

Figure 8:

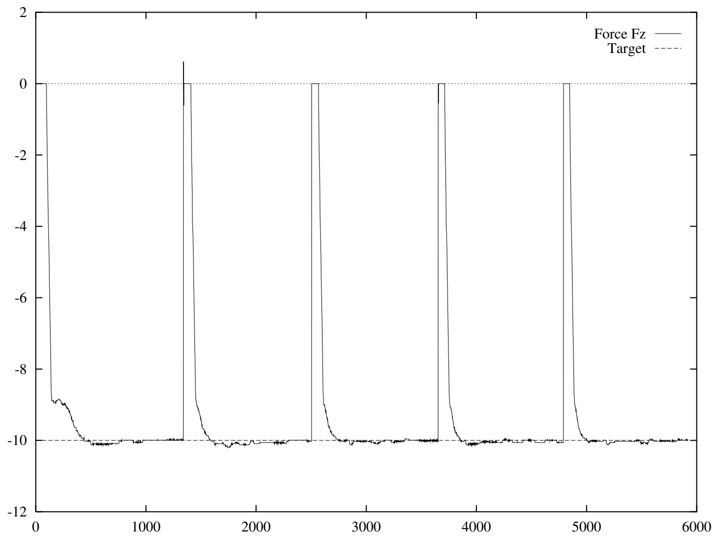


Figure 9:

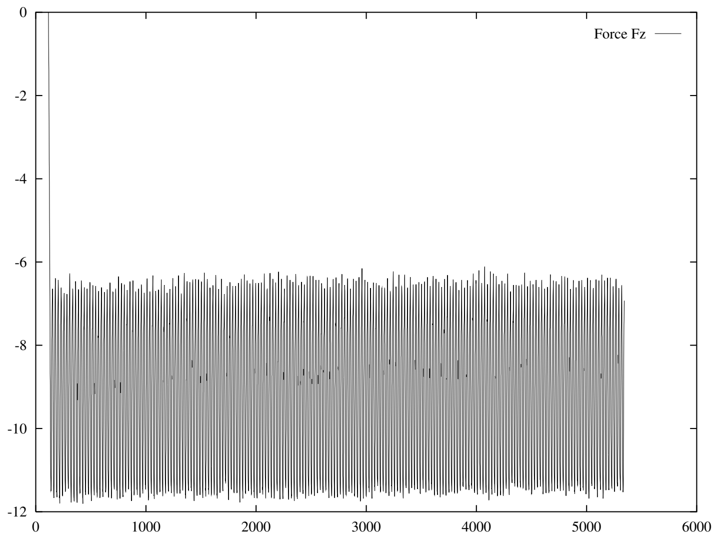
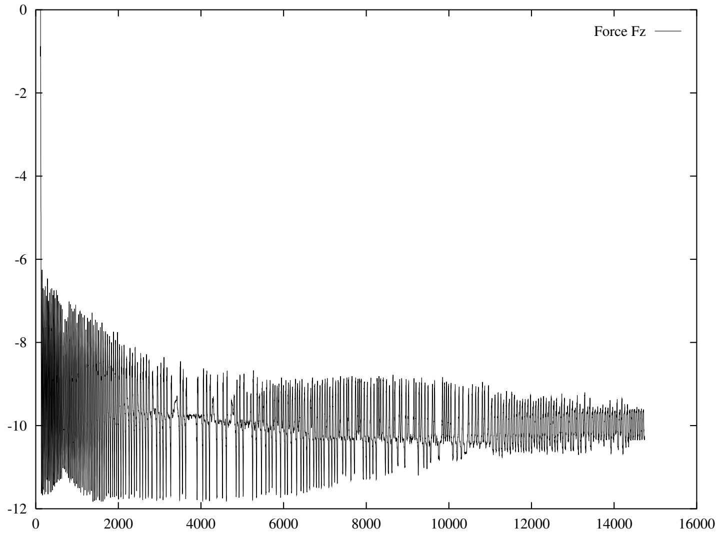


Figure 10:


Force Fz
0
-2
-4
-6
-8
-10
-12
0
2000
4000
6000
8000
10000
12000
14000
16000