From: AAAI Technical Report SS-96-02. Compilation copyright © 1996, AAAI (www.aaai.org). All rights reserved.

PBD – THE KEY TO SERVICE ROBOT PROGRAMMING

HOLGER FRIEDRICH, MICHAEL KAISER and RÜDIGER DILLMANN
Institute for Real-Time Computer Systems & Robotics (IPR)
University of Karlsruhe
D-76128 Karlsruhe,Germany
E-Mail: {friedric|kaiser|dillmann}@ira.uka.de.

Abstract. *Currently, an important topic of robotic research is the design and development of 'service robots'. These devices shall work autonomously or in cooperation with humans in a dynamic, relatively unstructured environment like households or offices. A crucial component of such systems is the programming interface. Programming methods have to be developed that enable an operator who is not a programming expert to easily instruct the robot.* Programming by Demonstration (PbD) *offers a way to accomplish this task. In this paper, we describe the results of two research projects that dealt with the application of* PbD *in the field of robotics. Furthermore, we outline the design methodology for an integrated system providing an interactive robot programming interface based on demonstrations and the analysis of user intentions.*

Key Words. Programming by Demonstration, Machine Learning, Robotics

INTRODUCTION

One of the major cost factors involved in robotic applications is the development of robot programs. Especially the use of advanced sensor systems and the existence of strong requirements with respect to the robot's flexibility ask for very skillful programmers and sophisticated programming environments. These programming skills may exist in industrial environments but they are certainly not available if the use of robots in a personal environment is considered. For opening the expected new, mainly consumer-oriented service robot market, it is therefore **essential** to develop techniques that allow untrained users to use such a personal service robot both safely and efficiently.

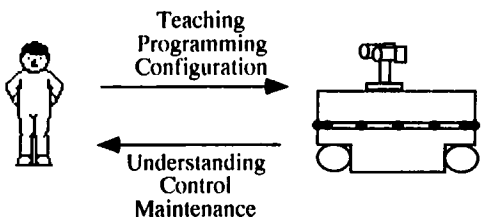

Figure 1: *Human-Robot Interaction in the context of personal service robots.*

Two basic aspects of the interaction between the robot and the user can be distinguished (Fig. 1). Firstly, the user wants to configure and instruct the robot. This requires to translate the user's language into the robot's, i.e., to compile user intentions into actual robot programs. Secondly, the low-level numeric representations used by the robot have to be translated into an understandable form, i.e., **symbols** have to be built from signals in order to allow the user to efficiently control and maintain the robot. These two aspects have to be taken into account to build robot systems that are able to acquire human action knowledge from observing human performance and to establish a suitable communication link to the user. For meeting these goals the *Programming by Demonstration (PbD)* paradigm (Cypher 1993) seems to be the right way to follow.
After an overview about the state-of-the-art in *Robot Programming by Demonstration (RPD)* and *Robot Skill Acquisition via Human Demonstration*, this paper does present the results of two research projects related to RPD and Learning in Robotics. Finally, the design methodology for a system that integrates the presented results in order to support *PbD* on different levels of robot control is given.

PbD in robotics – an overview

The *PbD* approach has been applied successfully in domains such as graphic editors (Lieberman 1993), instructable software agents (Maulsby 1994) or intelligent interfaces (Minton *et al.* 1995). Additionally, **Robot Programming by Demonstration (RPD)** has been realized through a number of applications on different levels of robot control.

- 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 that generated generalized programs by

fusing several demonstrations. Single demonstrations and user intentions are the basis for the construction of robot programs by the RPD system which is described later on in this paper.

- On the *control level*, demonstrations can be used as 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 dealing with the acquisition of closed-loop elementary skills in general feature a very task-specific design. They comprise acquisition techniques for manipulation tasks such as deburring (Asada and Liu 1991) and assembly (Dillmann *et al.* 1995a; Kaiser *et al.* 1995c) as well as for vehicle control (Pomerleau 1991) and autonomous robot navigation (Reignier *et al.* 1995).

Despite the increasing number of publications representing research in the various domains related to RPD, what is almost never taken into account is the *negative* influence the human teacher has on the learning system. Especially in a service environment with demonstrations provided by unexperienced users, examples must be considered to be not at all optimal with respect to the robot and/or the learning system. An analysis of the problems arising from the varying quality of user demonstrations in the robotics domain and ways to cope with these was presented in (Kaiser *et al.* 1995a).
Several projects that were concerned with the application of the *PbD* paradigm to robot programming were carried out at the IPR. One of those was the basic research project 'B-Learn II' which was funded by the European Union. Another project was performed in direct cooperation with SIEMENS. Results and conclusions derived from these projects will be presented in the following two sections.

The B-Learn II Project

The primary objective of B-Learn II was *the enhancement of robot programming techniques and robot autonomy by closing the loop between sensing and action through the introduction of learning capabilities.* This primary objective involved the achievement of several secondary ones, including

- the identification of learning tasks related to three case studies that represented important application fields in robotics,
- the assessment of the applicability of state-of-the-art machine learning techniques to solve these learning tasks,
- the identification of requirements that must be fulfilled by the robotic systems themselves, i.e., by both its hardware- and software-architecture in order to allow for learning to take place efficiently.

Besides other work the IPR's contribution to the project consisted of the development of skill acquisition techniques for manipulators and mobile systems, where skills requiring closed-loop control were of special interest.

Acquiring skills from human demonstration

On the **skill level**, control functions performing perception-action transformations must be learned from user demonstrations (Fig. 2) in order to actually realize a skill (Asada and Liu 1991; Kaiser *et al.* 1995c). An elementary skill is essentially given through a **control function**

$$C : \boldsymbol{u}(t) = C(\boldsymbol{y}(t-d), \ldots, \boldsymbol{y}(t-d-p)). \qquad (1)$$

with $\boldsymbol{u}$ being the control output and $\boldsymbol{y}$ being the sensory input. Given a sufficient amount of examples $((\boldsymbol{y}(t-d), \ldots, \boldsymbol{y}(t-d-p)), \boldsymbol{u}(t))$, C can be approximated through, e.g., a neural network, a set of fuzzy rules, or a regression tree. These techniques work well if the presented examples are in general "good," i.e., not contradictory and sufficiently distributed over the input space.
If the skill demonstration exhibited a constant correlation between inputs $\boldsymbol{y}$ and command $\boldsymbol{u}$, i.e., if the human teacher applied a constant strategy, d and p can be identified via correlation analysis, in order to generate training data from the demonstration data. If correlation analysis fails (which is often the case for real demonstration data), data analysis and training data generation becomes more complex and requires the following steps:

1. Identification of the relevant action components $\boldsymbol{u_i}$ based on the contribution of $\boldsymbol{u_i}$ to the overall motion involved in the demonstration.
2. Generation of a coarse model of the plant, i.e., determination of the signs of the partial derivatives $\frac{\partial \boldsymbol{y_j}}{\partial \boldsymbol{u_i}}$ on the base of the data recorded from the demonstration.
3. Identification of the relevant perception components using the results of step 1 and step 2.

If demonstrated solutions have been identified to be not optimal, it is not desirable to have the learning system simply copy them. The convergence criterion applied during the off-line learning phase must take care of this aspect. If the two boundary conditions

1. The direction of the learned action $\boldsymbol{u}^*$ must always be that of the demonstrated action $\boldsymbol{u}$, i.e., $\boldsymbol{u} = \alpha \boldsymbol{u}^*$ with $\alpha > 0$, and
2. The learned action $\boldsymbol{u}^*$ must not exceed the demonstrated one $\boldsymbol{u}$, i.e., $\boldsymbol{u} = \alpha \boldsymbol{u}^*$ with $\alpha \leq 1$

are considered, the criterion becomes

$$\boldsymbol{u} = \alpha \boldsymbol{u}^* \text{ with } \alpha \in (0, 1].$$

Obviously, the better the demonstration, the closer to 1 α can be.

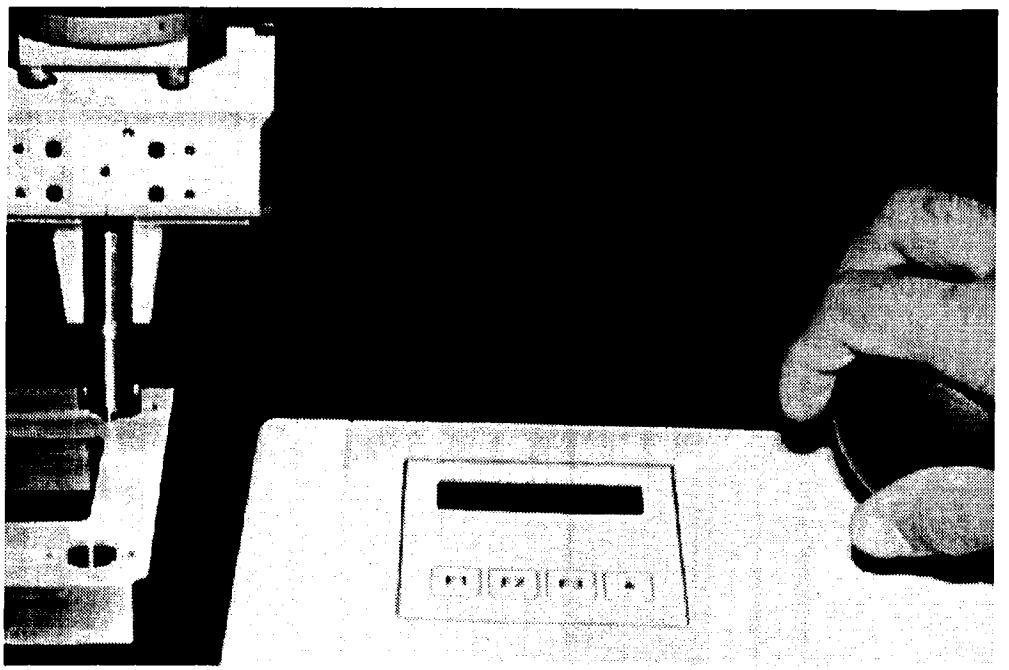

Figure 2: Demonstration of an insertion skill.

Lessons learned from B-Learn II

A common characteristics of all learning techniques that have been applied throughout B-Learn II is that they are basically **inductive** techniques. This is not astonishing if we consider the fact that - on the one hand - we did not really face any speedup learning problem, and - on the other hand - we did not want to provide a lot of formal background knowledge. However, a problem we always encountered was the **example generation.** The effort to provide examples to allow the application of inductive techniques was always significant.

Consequently, during the course of B-Learn II, one of the key points turned out to be the **interaction of the learning system with the user** (Friedrich 1995; Kaiser and Friedrich 1995). This also corresponds to the final focus of B-Learn II, which can shortly be described as **developing methods that allow for easier application and programming of robots especially for not production-oriented tasks** (Dillmann *et al.* 1995b; Kaiser *et al.* 1995b). If the realization of learning capabilities in robots should serve this purpose, the next step, i.e., to **ask** the human user for examples, feedback, etc. is obvious. In B-Learn II, we have made some important steps towards facilitating this kind of human-robot communication, and we have shown that the Programming by Human Demonstration paradigm, combined with sophisticated learning techniques, can actually enable robots to learn different things such as sensing and acting skills as well as communication and planning capabilities from humans.

The RPD Project

Within the RPD project, which was carried out in collaboration with the SIEMENS company the task was to generate structured task-level robot programs from user demonstrations.
The project itself ran over a timespan of $2\frac{1}{2}$ years. The prepared prototype had to meet the following demands. Firstly, it had to be able to run using a real PUMA260 robot arm. This required the development of a Robot-Control Tool (see fig. 4 left). This tool provides an interface that allows the user to control the manipulator directly, to choose other control devices (6D-spacemaster), and to configure data recording parameters.
Secondly, it had to be granted that the system would always generate programs that contain exclusively user intended generalizations. The architecture displayed in figure 3 was chosen as basis of the system. It is human centered and integrates the user in a way that allows him to control, support and supervise the activities of each and every system component.
Besides others two main tasks that had to be solved in order to fulfill the objectives outlined above arised in the course of the project.

Task1: A demonstration is recorded and sampled as a sequence of position vectors of the manipulator. The task was to transform this subsymbolic representation into a symbolic one that is interpretable by the user as well as the learning modules.

Task2: Given a symbolic representation of a demonstration, the task was to develop an algorithm that processes the data in order to generate *user intended generalized plans* that are applicable to a whole class

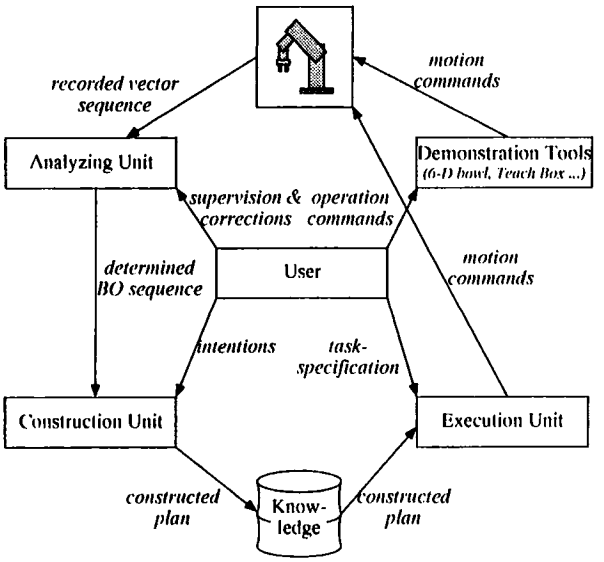

Figure 3: *The human centered architecture of the RPD system.*

of tasks, that can be transformed into robot programs and finally be executed on a real robot device.

Vector Sequences to Symbolic Operators – a robust transformation

According to the first task described above, an *analysis* of the vector trace representing a demonstration is necessary to transform the "raw demonstration data" into the symbolic format of the planning language. The chosen planning language consists of STRIPS-like Operators (Fikes *et al.* 1972) that are *Basic Operators (BOs)* (e.g. transfer, grip, approach, ...) and instructions describing branches and loops.

A single demonstration is firstly transformed into a sequence of *BOs* that is developed into a more complex structure by several subsequent learning modules. The recognition of the sequence of BOs hidden in the recorded vector sequence (i.e., a correct transformation), is a twofold problem. The trace has to be segmented into subsequences and each of these has to be classified and mapped on one of the five BOs.

The two tasks of *segmentation* and *classification* depend on each other and therefore have to be done in parallel. In this sense, the transformation task is similar to the problem of continuous speech recognition. Another aspect also common to the speech recognition domain is that of time variations in the signal(the demonstration). In the domain of speech recognition, *Time Delay Neural Networks (TDNNs)* have been proven to be powerful tools for various segmentation and classification problems (Waibel *et al.* 1989), overcoming also the problem of time variations in the input signal.

Therefore, the application of TDNNs was considered for the solution of the analysis problem in the RPD system. The demonstration's vector sequence is fed simultaneously into several TDNNs (one for each BO). By studying the networks' outputs, a decision about which BO was performed in a certain part of the demonstration is taken. Thereafter the determined symbolic operator sequence is instantiated.

Up to this point the transformation process is fully automatic. But what if misclassifications occur and thus the constructed symbolic operator sequence does not represent the given demonstration? Since misclassifications in the transformation process might have negative effects on following generalisation steps, the user must have the opportunity to supervise and, if necessary correct the result of the automatic classification and transformation process. Therefore, manual correction of classification errors and supervision of the analysis module's results, a graphical user interface is provided (see fig. 4 .right). The manipulator's trajectory is plotted and the boundaries of the recognized BOs are given. Now the user has the possibility to correct the operator sequence in order to improve the classification results. Following the transformation to the symbolic operator level, the generalization and plan generation takes place. This requires solving the second task stated above. A solution will be given in the next paragraph.

Acquiring User Intentions – the key to directed generalization

One problem which frequently occurs when employing current machine learning algorithms is that in some cases they learn concepts or classifiers that are consistent with the available data, background knowledge and reasoning mechanism but nevertheless do not reflect the result the user desires.

Since robotic devices are potentially harmful for the environment they do act in, robot programming is a field where the user expects a system to learn and do exactly what he or she wants it to learn and do. Therefore, the programming/learning system has to generate/generalise program schemata that represent the user's intentions. Moreover, this goal should be achieved by simply analyzing a single demonstration, since having to demonstrate similar tasks multiple times with a robot would never be acceptable for an average user. To accomplish this, several possible incidents have to be taken into account. Not all actions within a demonstration may reflect the user's intention exactly. Some actions may simply be forced by the environmental conditions present at that moment. Some of the manipulations may even be unnecessary in order to solve the given task.

Unfortunately, a computer system is not able to *guess* whether one of these incidents occurred in a demonstration or what the user's intentions were. Since the primary focus in the developed RPD system was not to induce generalizations and/or knowledge from a large number of examples, a different direction than classical ML induction techniques had to be followed to generate the desired answers. In order to keep the number of required demonstrations as small as possible, another source of information had to be integrated in the induction process–the user itself. Therefore, the user

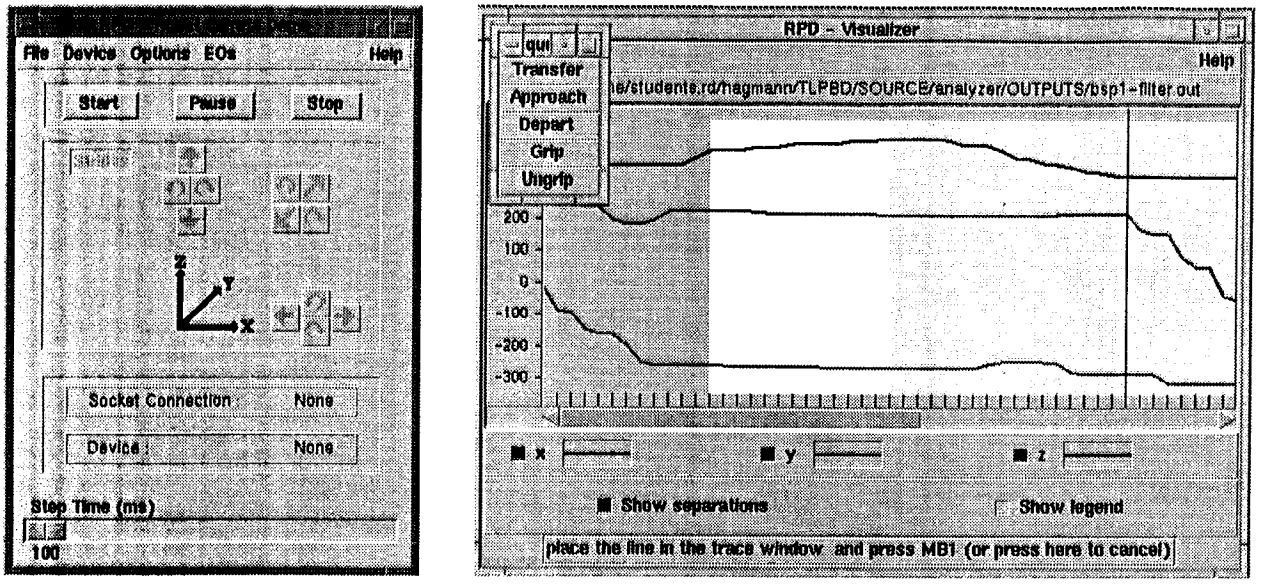

Figure 4: *Left: User interface for robot control, demonstration, configuration of data recording; Right: trace with classified segments. The border of a missing segment is about to be inserted.*

was consequently embedded as an active provider of knowledge into the learning process. This gives him the possibility (and duty) to explain parts of a demonstration that weren't interpretable for the system and to correct errors that he may have made in the course of the demo.

At first, the system processes the symbolic operator sequence representing the demonstration and calculates pre- and postconditions of all operators, subsequences and the whole demonstration itself. Then all actions and their effects are known. Thereafter, the system asks the user for his comments and intentions. Having perceived this additional knowledge, the BO sequence is processed again, branches are inserted if possible and unnecessary actions are erased. Thus the user intended generalization takes place. The process of fusing a single demonstration with the users intentions is described in great detail in (Friedrich and Dillmann 1995).

A methodology for robust transfer of human knowledge employing PbD

To combine the results achieved in the previously described projects, a system integrating the developed techniques is currently under construction. The applied design methodology is described below.

Since one of the main characteristics of knowledge provided via human demonstrations of actions is the varying quality of this knowledge, several issues become very important. These are

- example quality assessment,
- on-line refinement of the initially acquired knowledge, and
- an appropriate man-machine interface that provides features for easy robot control as well as for robust and comfortable interaction between the user and the learning system.

For acquired elementary skills, their immediately available operational representation (e.g., as a neural network) must be complemented by a symbolic one, in order to make this new knowledge accessible for both the human teacher and the robot's reasoning mechanism and to include the acquired skills into symbolic task-level programs.

Thus, the knowledge transfer process (Fig.5) becomes complex, requires more steps than the usually considered example generation, "strategy extraction" (learning), and skill application, and asks for the support of the human teacher in several phases:

1. The teacher determines tasks for the robot that require to extend the robot's knowledge base, i.e. to acquire a new task program schema or elementary skill.
2. The teacher selects the scenario for the generation of the examples and performs a demonstration, either a manipulation sequence for a task-level program acquisition or an elementary skill. In the case of skill acquisition, a strategy for autonomous experimentation including boundary conditions on perceptions and actions and an evaluation function can be specified alternatively. Demonstrations for task level programs are interrupted to demonstrate specific skills if these aren't present in the knowledge base yet.
3. The teacher, supported by analysis tools, assesses the quality of the example, thereby guiding both the preprocessing and the definition of convergence criteria.
4. If an elementary skill has been acquired, the teacher adds a symbolic interpretation, supported by the learning robot that provides the context in which the

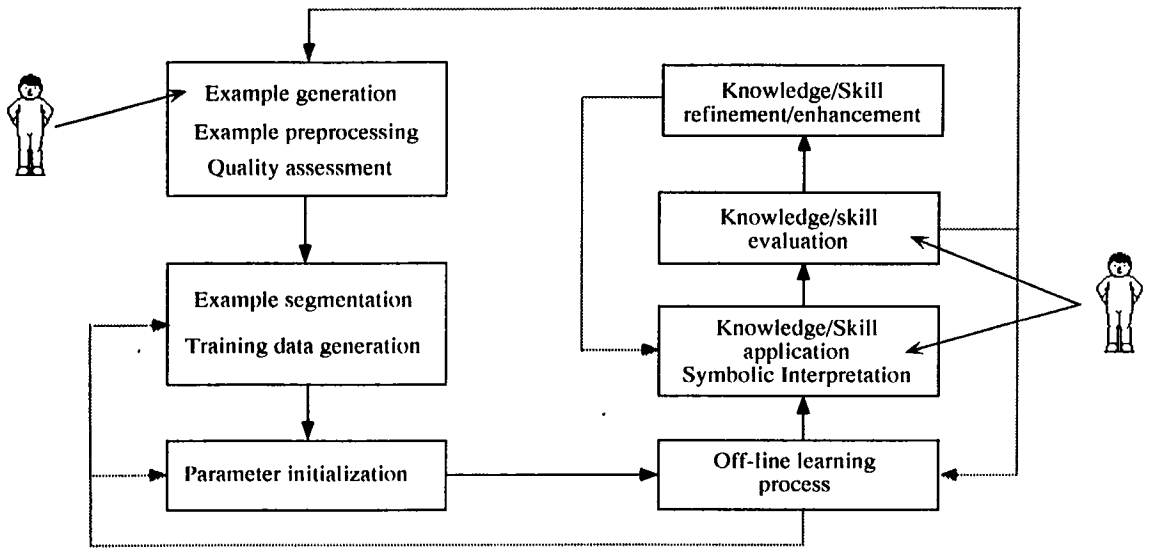

Figure 5: Different phases of user supported program knowledge/skill acquisition, application, and refinement. Gray arrows indicate feedback loops. Arrows pointing from the users indicate user interactive phases.

missing skill was detected (originating from step 1). Thus the skill's subsymbolic representation, which is operational for the low-level robot control systems is embedded in a symbolic one in order to provide operationality of the skill for the task-level *PbD*, planning, and control components.

5. If a whole manipulation sequence has been acquired, the teacher supplies his intentions to guide the generalization process of the symbolic task-level learning modules in the intended direction.
6. The teacher provides an on-line evaluation of the robot's performance, either by subjectively evaluating it, by choosing a general evaluation function (e.g., based on motion/recognition speed), or by designing an evaluation function specific for the newly acquired skill, or program schema.

The human teacher is therefore heavily involved in the robot's learning process. However, the cumbersome task to program the robot has been replaced by actually *communicating* with the robot either on a high level of abstraction or via *showing* solutions instead of formally specifying them.

DISCUSSION

In this paper we presented the results of two research projects that were concerned with the application of *PbD* to robotics. We showed that to rely on the ability of a human teacher to demonstrate a solution to a given task, and to provide at least a qualitatively correct evaluation of the robot's performance, is not only realistic but necessary for easy programming of service robots.

We cannot expect that the elementary action skills and task-level programs acquired via an interactive learning approach are comparable to those originating from an in-depth task analysis and explicit robot programming. It is obvious that in order to raise the quality and applicability of skills and program schemata obtained from demonstrations further research effort has to be spend in the areas of human computer interaction, machine learning, and sensor data processing.

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, i.e., Robot Programming by Demonstration, definitely is.

ACKNOWLEDGEMENTS

We'd like to thank the project partners involved in B-Learn II and our colleagues at SIEMENS for their help and cooperation. This work has partially been supported by the ESPRIT Project 7274 "B-Learn II". It has been performed at the Institute for Real-Time Computer Systems & Robotics, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe, Germany.

References

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

H. Asada and S. Liu. Transfer of human skills to neural net robot controllers. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991.

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

N. Delson and H. West. The use of human inconsistency in improving 3D robot trajectories. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 1994.

R. Dillmann, M. Kaiser, and A. Ude. Acquisition of elementary robot skills from human demonstration. In *International Symposium on Intelligent Robotics Systems*, Pisa, Italy, 1995.

R. Dillmann, M. Kaiser, F. Wallner, and P. Weckesser. PRIAMOS: An advanced mobile system for service, inspection, and surveillance tasks. In T. Kanade H. Bunke, H. Noltemeier, editor, *Modelling and Planning for Sensor Based Intelligent Robot Systems*. World Scientific, 1995.

R.E. Fikes, P.E. Hart, and N.J. Nilsson. Learning and executing generalized robot plans. *Artificial Intelligence*, 3(4), 1972.

H. Friedrich and R. Dillmann. Robot programming using user intentions and a single demonstration. In M. Kaiser, editor, *Proceedings of the 3rd European Workshop on Learning Robots (EWLR-3)*, Heraklion, Crete, Greece, April 1995.

H. Friedrich, editor. *Learning from Examples versus Programming by Demonstration*, Tahoe City, California. 1995. 12th International Conference on Machine Learning.

R. Heise. *Demonstration instead of programming: Focussing attention in robot task acquisition*. Research report no. 89/360/22, Department of Computer Science, University of Calgary, 1989.

M. Kaiser and H. Friedrich. Transferring human knowledge to robots. In *Meeting GI Special Interest Group Machine Learning*, Dortmund, Germany, 1995. Also available via anonymous ftp from ftpipr.ira.uka.de.

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

M. Kaiser, V. Klingspor, J. del R. Millàn, M. Accame, F. Wallner, and R. Dillmann. Using machine learning techniques in real-world mobile robots. *IEEE Expert*, 1995.

M. Kaiser, A. Retey, and R. Dillmann. Robot skill acquisition via human demonstration. In *Proceedings of the International Conference on Advanced Robotics (ICAR '95)*, 1995.

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*, 10(6):799–822, 1994.

Henry Lieberman. *Watch what I do*, chapter MONDRIAN: A teachable graphical editor. 1993.

Dave Maulsby. *Instructible agents*. PhD thesis, University of Calgary, Canada, 1994.

S. Minton, A. Philpot, and S. Wolfe. Specification by demonstration: The ViCCS interface. In *International Conference on Machine Learning, Workshop on Programming by Demonstration*, Tahoe City, California, 1995.

D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3:88 – 97, 1991.

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

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

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

A. Ude. Trajectory generation from noisy positions of object features for teaching robot paths. *Robotics and Autonomous Systems*, 11(2):113–127, September 1993.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on acoustics, speech and signal processing*, pages 328 – 339, March 1989.