TEACHING SERVICE ROBOTS COMPLEX TASKS: PROGRAMMING BY DEMONSTRATION FOR WORKSHOP AND HOUSEHOLD ENVIRONMENTS

Markus Ehrenmann, Oliver Rogalla, Raoul Zöllner and Rüdiger Dillmann
Industrial Applications of Informatics and Microsystems, Haid-und-Neu-Str. 7,
D-76131 Karlsruhe, Universität Karlsuhe (TH), Germany
Phone +49(0)721-608-3547, Fax +49(0)721-608-8270,
E-mail: [ehrenman|rogalla|zoellner|dillmann]@ira.uka.de

Abstract: Programming of service robots is an expensive and difficult task especially when manipulator arms are involved. This is one of the drawbacks for every day use of these systems. *Programming by Demonstration* is a means to let users program robots simply by demonstrating a task like putting a table or composing an object to a system that observes, interprets and then maps the performed user action to a given manipulator. At our institute, observation of a task is realised using active vision systems, a data glove and magnetic field sensors. After the demonstration, the system interprets and stores the recorded actions segmenting them into meaningful parts like grasps of objects or sticking objects to others. Due to sensor errors and the complexity of the problems, the system puts queries concerning grasp types or object positions. A robot should then be able to perform the same actions in a likewise environment. So far, execution has shown good results in simulation. An appropriate service robot is currently being set up at our institute in order to prove the feasibility of our approach. It is equipped with a stereo camera head and a seven DOF manipulator arm with a three finger gripper.

Keywords: Service robots, Learning, Sensor Fusion, Programming by Demonstration

1 INTRODUCTION

Programming service robots is still an annoying task. Above all, manipulator arms require a lot of expert knowledge. Every unit needs to be installed and prepared for it's particular environment. Thus, use of these systems is still an expensive and long lasting venture. The expected consumer market for many-purpose service systems will neither accept today's user interfaces nor today's programming techniques. As we believe, *Programming by Demonstration (PbD)* is a technique that overcomes the drawbacks of classical approaches and could help let people program robots easily. Only this will make them buy their own systems. The idea is to have a human demonstrate a task solution. The demonstration is observed, recorded and interpreted. Then, manipulator kinematics can be defined that the movement traces and actions are mapped to. The formerly observed actions can now be executed on a robot target system with appropriate controllers. These controllers could be supplied by the manufacturers in the future.

At our institute, experiments following the *PbD* paradigm have been undertaken in a fixed environment. In this paper, an overview about today's *PbD* approaches will be given in section 2. Section 3 introduces a framework for certain *PbD* phases and their challenges. Furthermore, the structure of our system solving these challenges is outlined. The 4th section describes the particular functionality of the modules of our system in the previously identified *PbD* phases. After presenting future work in section 5, the paper closes in section 6.

2 STATE OF THE ART

Several *PbD* systems and approaches have been proposed during the past years. An overview and classification can be found in [Dillmann et al., 99]. Learning of action sequencies or operation plans is an abstract problem, which supposes a modelling of cognitive skills. When learning complex action sequencies, basic manipulation skills or controlling techniques become nonrelevant. In fact, the aim is to generate an abstract description of the demonstration reflecting the user's intention and modelling the problem solution preferably optimally. Likewise, a given problem has to be suitably generalised, for example distinguishing parameters specific for the particular demonstration and parameters specific for the problem concept. Both demands a certain insight in the environment and the user's performance.

Often, the analysis of a demonstration takes place observing the changes in the scene undertaken by the user. These changes can be described using relational expressions or contact relations [Kuniyoshi 94, Ikeuchi 93, Onda 97]. For generalising of a single demonstration mainly explanation based methods are used [Mitchel 86, Friedrich 99]. Those allow for an adequate generalisation taken from only one example (*One-Shot-Learning*).

A physical demonstration in the real world can be too time-consuming and may not be mandatory.

That is why some researchers rely on virtual or iconic demonstrations [Archibald 93]. The advantage of performances in a virtual world is that here manipulations can be executed that would not be feasible in the real world because of physical constraints. Furthermore, inaccuracies of sensor based approaches do not have to be taken into account. This way, in existing approaches object poses or trajectories [Takahashi 96, Tung 95, Kang 97] and/or object contexts [Onda 97, Heise 92, Segre 89, Friedrich 96] are generalised in an appropriate way.

Iconic programming starts at an even more abstract level. Here, a user may fall back on a pool of existing or acquired basic skills. These can be retrieved through specific cues (like icons) and embedded in an action sequence. The result of such a demonstration of operator sequences can then be abstracted, generalised and summarised as new knowledge for the system. An example can be found in the skill-oriented programming system SKORP [Archibald 93] that helps putting up macro operators out of actuator or cognitive elementary operators interactively.

Besides a derivation of action sequences from a user demonstration direct cooperation with users has been investigated. Here, user and robot reside in a common work cell. The user may direct the robot on the basis of a common vocabulary like speech or gestures [Zhang 99, Voyles 99, Steinhage 98]. But this allows for rudimentary teaching procedures only.

3 *PBD* FOR MANIPULATION TASKS

Our approach to *PbD* emphasizes the interpretation of what has been done by the human demonstrator. Only this enables a system to reuse formerly observed action sequences in different environment. It will be shown that the process of learning complex problem solving knowledge for robot systems consists of different phases.

3.1 THE *PbD* PROCESS

Like summarized in figure 1, the whole programming process starts with a user demonstration of a specific task which is observed by a sensor system. The following phases seem to be the basic components for making successfully use of data from a human demonstration:

1. Sensor systems are used for observing the users movements and actions. Also important changes like object positions and constraints in the environment can be detected. The sensor efficiency might be improved by allowing the user to comment his actions.
2. During the next phase relevant operations or environment states based on the sensor

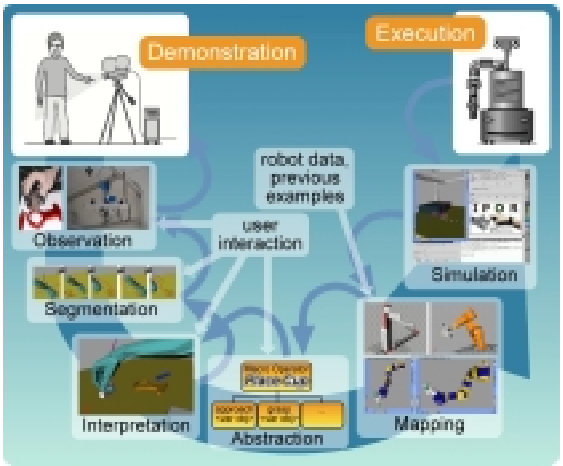

Figure 1: PbD Process

as possible. Generalization of the obtained operators includes further advice by the user. Spontaneous and not goal oriented motions may be identified and filtered in this phase. It is considerable to store the task knowledge in a form that it is reuseable even if execution conditions do vary slightly from the demonstration conditions

5. Transfer of the internal knowledge representation to the target system. As input serves the generated task solution knowledge from the previous phase. Additionally background knowledge about the kinematic structure of the target system is required. Within this phase as much information as possible available from the user demonstration should be used to allow robot programming for new tasks.
6. In the simulation step the generated robot program is tested against it's applicability in the execution environment. It is also desirable to allow the user to confirm correctness to avoid dangerous situations in the execution case.
7. During execution success and failure can serve for modifications on the current mapping strategy (e.g. if a picked workpiece slips out of the robot's gripper the higher grasping forces could be selected)

The overall process with its identified phases is suitable for manipulations tasks in general. However, in practice the working components are restricted to a certain domain to reduce environment and action space.

Within our research we concentrated on pick-and-place operations first, since these are basic prerequisites for many service tasks. However, we experimented with real household objects like cups and plates and support different grasp-types for distinct robots.

3.2 SYSTEM STRUCTURE

To meet the requirements for each phase of the PbD-process identified in the previous section we have developed a system consisting of the components shown in figure 2.

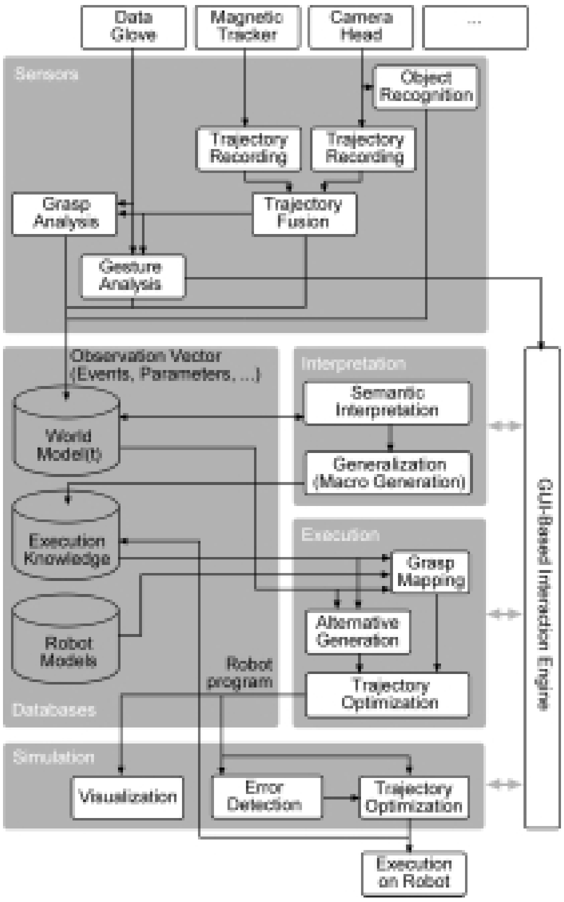

Figure 2: System structure

The system integrates four basic modules. The sensor module is responsible for analysis and presegmentation of information channels giving connection to the sensors. It's output is a vector describing states in the environment and user actions. This information is stored in a database including the world model.

The second module operates on the gained observation vectors and associates sequences of observation vectors to a set of predefined symbols. These parametrizable symbols do represent the elementary action set. During the interpretation phase the symbols are chunked in hierarchical macro operators after replacing specific task-dependant parameters by variables. The result is stored in a database as generalized execution knowledge.

This knowledge is taken from the execution module which uses specific kinematic robot data for processing. It calculates optimized movements for the target system taking into account the actual world model.

Before sending the generated program to the target system it's validity is tested throughout a simulation. In case of unforeseen errors, movements of the robot have to be corrected and optimized.

All four components do communicate with the user by a graphical user-interface. Additional information can be retrieved or hypotheses can be accepted or rejected. While demonstrating, the user may use gestures for interaction as well.

4 *PBD* PHASES

In the current implementation, every phase of the *PbD* process is supported as follows (numbering refers to the phases mentioned in section 3):

Figure 3: Trinocular active camera head

2. During the demonstration process, the user handles objects in a training center. This is equipped with the following sensors: a data glove, a magnetic field based tracking system and an active trinocular camera head (see figure 3). Object recognition is done by computer vision approaches using fast view-based approaches described in [Ehrenmann et al. 99]. From the data glove, the system extracts finger joint movements and hand positions in 3D space. To reduce noise in trajectory information, the user's hand is additionally observed by the camera system. Both measurements are fused using confidence factors, see [Ehrenmann et al. 01b]. Furthermore, pretrained neural networks evaluate hand configurations to detect certain events like grasps and gestures. The system is able to distinguish 16 different grasp types and 9 distinct gestures (see [Friedrich et al. 99]). This information is stored with discrete time stamps in the world model database.
3. For interpretation, the system employs two types of predefined symbols: first, 16 different grasp types following the Cutkosky hierarchy (see [Cutkosky 89])

and second, three types of basic movements (linear, free and spline move). The data from the demonstration (retrieved from the database) is now analyzed by an internal heuristic: a grasp is performed when three conditions hold:

   a. neural detectors fire,
   b. the user's hand speed is below a certain threshold
   c. the fingers are close to a manipulable object

   The move types are detected by analyzing speed and trajectory type of the user's hand. A free move is detected when the trajectory leaves an enclosing cylinder while a linear move stays inside. A spline move is selected when a free move can be associated to a simple spline curve. The thresholds and parameters can be modified by the user if necessary. Result of this phase is a sequence of the proposed symbol set associated including important data recorded during demonstration. For instance, finger poses and positions for grasps, particular objects or positions in 3D space.

4. The symbol sequence from the previous phase is now chunked in semantically related groups. These are for example: approach phases, grasp and ungrasp phases. This process continues recursively controlled by an internal heuristic. The heuristic is designed by analysis of pick-and-place operations. For instance, a macro operator for a simple pick-and-place operation has a tree-like structure starting from the whole operation at the root. The first branches lead to a symbol for a *pick* and a *place*. They are subdivided in *approach*, *grasp/ungrasp* and *disapproach* phases. Finally, the process ends up with the elementary operations at the leaves.

   Each component in that macro structure has certain context conditions like preconditions. They are generated automatically from the demonstration (see [Friedrich et al. 98]). Context conditions decide whether a macro operator is applicable for a certain environment situation. It is important to understand that all trajectory parameters stored inside a macro is object-dependant. For instance, during an approach operation all movements are represented relatively to the object of interest. This makes it possible to substitute specific objects by variables. If the macro is re-used again, variables become replaced by concrete objects that match context conditions.

5. Up to now, problem solving information stored in the macro operators is not directly usable for the target system. Grasps are represented in order to describe human hand configurations and trajectories are not optimal for robot kinematics. Besides, sensor control for the robot like f.e. force control by force/torque sensors is not extractable from the demonstration.

   Since we are dealing with pick-and-place operations, we have developed a method for automatic mapping of grasp types to robot grippers. Furthermore, stored trajectories in the macro get trimmed for the execution environment and the target system. The system uses generic robot and gripper models described in [Rogalla et al. 00]. This makes it possible to support different gripper and manipulator types for execution. (see figure 4).

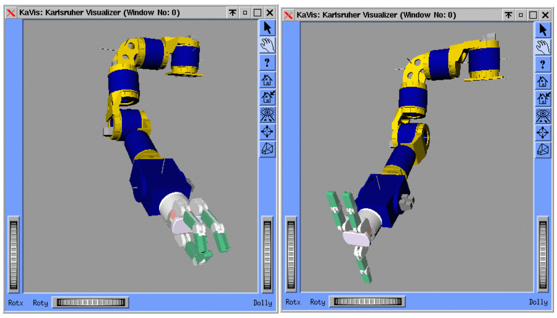

Figure 4: Robot arm with three finger gripper

   Additionally, we defined a set of logical rules that select sensor constraints depending on the execution context. This is for example to select a force threshold parallel to the movement when approaching an object.

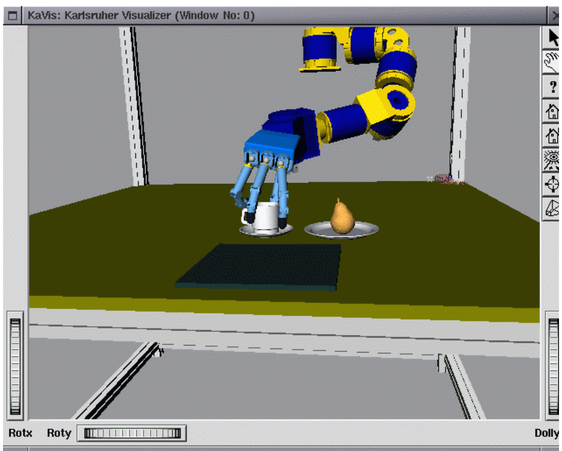

Figure 5: Simulation of generated program

6. The generated robot program is simulated in a graphical environment. Here, the user

might reject certain actions and movements (confer figure 5).

5 EXPERIMENTS

As programming task, we addressed table laying among others as an example to demonstrate feasibility of our approach. For this purpose, we selected real-world cups and plates in different forms and colors. The examples were recorded in our training center (see figure 6). The mapping was shown with different robot types in simulation. For execution, our new service robot ALBERT will be utilized.

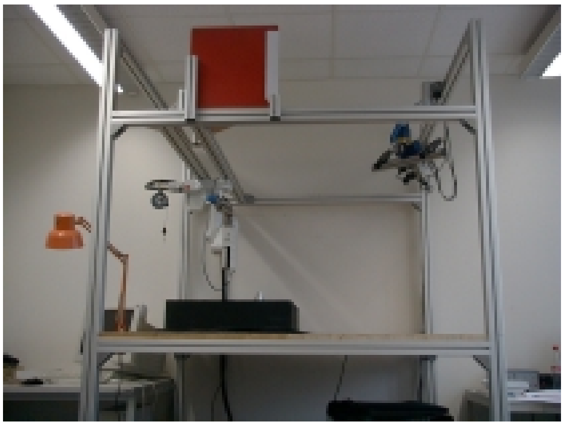

Figure 6: Training center with sensors

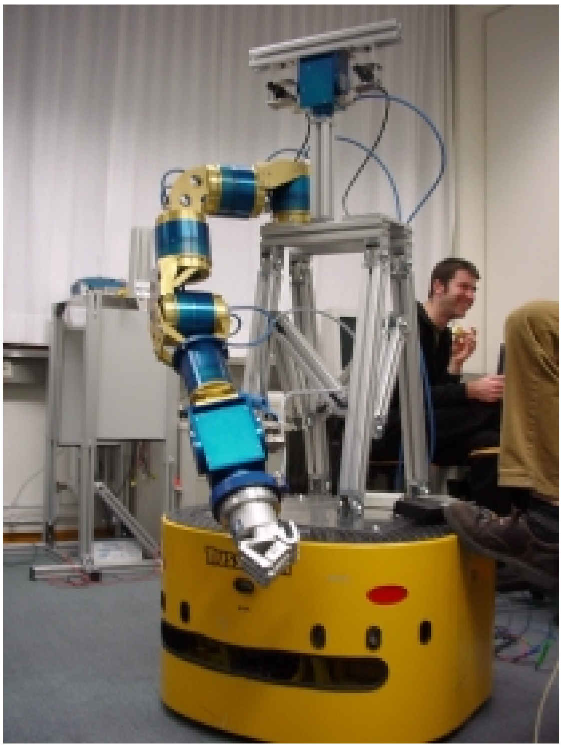

Figure 7: Service robot ALBERT

The robot is equipped with a stereo camera head and a 7 DOF light-weight arm which was shipped by Amtec, Berlin (see figure 7). The arm weighs about 35kg and can lift objects of up to 10kg with fully extended modules. A 6 DOF force/torque sensor supplied by the DLR, Munich interconnects the arm with the three finger Barrett hand. The upper body part is mounted on a mobile platform ODETE which was developed at our institute. ODETE is equipped with supersonic sensors and a planar SICK laserscanner for self-localization and obstacle recognition [R. Graf and P. Weckesser 98]. With it's differential drive, the platform will enable the robot to move freely in a workshop or household environment.

Since all robot functions are not implemented yet (force control of the arm), experiments are not yet finished.

6 FUTURE WORK

For demonstration of a task to the service robot, the user interface will enhanced. Several interaction channels will allow for intuitive interaction between man and machine:

- Speech recognition will be realized using IBM's ViaVoice. So graphical dialogues for user comments will be substituted.
- Gesture recognition considering static and dynamic gestures. A first approach has already been tested successfully [Ehrenmann et al. 01a]. Gestures will be used for commanding, commenting on a demonstration and responding in user dialogues.
- Beside pick-and-place operations, actions like dynamic grasps where objects are manipulated with the fingers during grasping will be included in the system.

7 CONCLUSION

PbD has shown good results programming manipulator arms giving users the ability to create general robot programs without having specific knowledge about kinematics. Having tested this approach in a training center and in simulation, now the programming system will be used for program generation for the service robot.

8 ACKNOWLEDGEMENT

This work has partially been supported by the DFG project "Programmieren durch Vormachen" and mainly by the BMBF project "Morpha".

9 REFERENCES

[Archibald et al 93] C. Archibald and E. Petriu. Computational paradigm for creating and executing sensorbased robot-based programming. ISIR, 401-406, Tokyo, Japan, 1993.

[Cutkosky 89] M. R. Cutkosky. On Grasp Choice, Grasp Models, and the Design of Hands for

Manufacturing Tasks. IEEE Transactions on Robotics and Automation, 5(3), 269-279, 1989

[Dillmann et al. 99] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner and M. Bordegoni. Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm. 9th International Symposium of Robotics Research (ISRR '99), 229-238, Snowbird, Utah, USA, October 9-12, 1999

[Ehrenmann et al. 00] M. Ehrenmann, D. Ambela, P. Steinhaus and R. Dillmann. A Comparison of Four Fast Vision Based Object Recognition Methods for Programing by Demonstration Applications, Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA), San Francisco, California, USA, 1862-1867, April 2000

[Ehrenmann et al. 01a] M. Ehrenmann, T. Lütticke and R. Dillmann. Dynamic Gestures as an Input Device for mobile platforms. IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21-26, 2001 (to be published)

[Ehrenmann et al. 01b] M. Ehrenmann, R. Zöllner, S. Knoop and R. Dillmann. Sensor Fusion Approaches for Observation of User Actions in Programing by Demonstration. Proceedings of the IEEE International Conference on Multi Sensor Fusion and Integration (MFI), Baden-Baden, Germany, August 2001 (to be published)

[Ehrenmann et al. 99] M. Ehrenmann, P. Steinhaus and R. Dillmann, A Multisensor System for Observation of User Actions in Programing by Demonstration, Proceedings of the IEEE International Conference on Multi Sensor Fusion and Integration (MFI), Taipeh, Taiwan, 153-158, August 1999

[Friedrich et al. 96] H. Friedrich, S. Münch, R. Dillmann, S. Bocionek and M. Sassin. Robot programming by demonstration: Supporting the induction by human interaction. Machine Learning, Seiten 163-189, Mai/Juni 1996.

[Friedrich et al. 98] H. Friedrich, O. Rogalla and R. Dillmann. Interactive robot programming based on human demonstration and advice. H. Christensen (ed.): Dagstuhl Proceedings 1998, Springer, Dagstuhl, Germany

[Friedrich et al. 99] H. Friedrich, V. Grossmann, M. Ehrenmann, O. Rogalla, R. Zöllner and R. Dillmann. Towards cognitive elementary operators: grasp classification using neural network classifiers, Proceedings of the IASTED International Conference on Intelligent Systems and Control (ISC), Santa Barbara, California, USA, Octobre 1999

[R. Graf and P. Weckesser 98] R. Graf and P. Weckesser. Roomservice in a hotel. 3rd IFAC Symposium on Intelligent Autonomous Vehicles – IAV 98, 641-647, Madrid, Spain, March 1998

[Kang et al 97] S. B. Kang, K. Ikeuchi. Toward automatic robot instruction from perception – mapping human grasps to manipulator grasps. IEEE Transactions on Robotics and Automation, 13(1), Februar 1997

[Mitchel 86] T.M.Mitchell. Explanation-based Generalization - a unifying View. Machine Learning, 1;47-80, 1986.

[Onda et al 97] H. Onda, H. Hirukawa, F. Tomita, T. Suehiro, and K. Takase. Assembly motion teaching system using position/force simulator – generating control program. 10th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS`97), Grenoble, Frankreich, September, 7-11, 1997.

[Rogalla et al. 00] O. Rogalla, K. Pohl and R. Dillmann. A General Approach for Modelling Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems. Takamatsu, Japan, October 30-November 5, 2000

[Segre 89] A.M. Segre. Machine Learning of Assembly plans. Kluwer Academinc Publishers, 1989

[Steinhage et al 98] A. Steinhage, T. Bergener. Dynamical Systems for the Behavioral Organization of an Anthropomorphic Mobile Robot. In R. Pfeifer, B. Blumbert, J.A. Meyer and S.W. Wilson (Editors), From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB 98). MIT Press, August 1998.

[Takahashi et al 96] T. Takahashi. Time normalization and analysis method in robot progrmming from human demonstration data. Proceedings of the IEEE International Conference on Robotics and Automation, 695-700, Atlanta, Georgia, USA, 1993

[Tung et al 95] C.P.Tung, A.C. Kak. Integrating sensing, task planning and execution. IEEE International Conference on Robotics and Automation, 3;2030-2037, 1994

[Voyles 99] R. Voyles and P. Khosla, Gesture-Based Programming: A Preliminary Demonstration. Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, Michigan, 708-713, May 1999

[Zhang et al 99] Jianwei Zhang, Yorck von Collani, Alois Knoll. Interactive Assembly by a Two-Arm-Robot Agent. Robotics and Autonomous Systems, Elsevier 1999.

[Zöllner et al. 01] R. Zöllner, O. Rogalla and R. Dillmann. Integration of Tactile Sensors in a Programming by Demonstration system. IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21-26, 2001 (to be accepted)