A Two-Arm Situated Artificial Communicator for Human-Robot Cooperative Assembly [1]

Jianwei Zhang
Department of Computer Science,
University of Hamburg,
22527 Hamburg, Germany
zhang@informatik.uni-hamburg.de

Alois Knoll
Department of Computer Science,
Technical University of Munich,
81667 Munich, Germany
knoll@informatik.tu-muenchen.de

Abstract

We present the development of a robot system with advanced cognitive capabilities. We focus on two topics: assembly by two hands and understanding human instructions in natural language with few constraints. A typical application of such a system is human-robot cooperative assembly. A human communicator sharing the view of the assembly scenario with the robot instructs the latter by speaking to it in the same way that he would communicate with a child whose common sense knowledge and motor skills are limited. His instructions can be under-specified, incomplete and/or context-dependent.

After introducing the general purpose of this research project, we present the hardware and software components of our robots needed for interactive assembly tasks. We then discuss the control architecture of the robot system with two stationary robot arms by describing the functionalities of perception, instruction understanding and execution. To show how our robot learns from humans, the implementations of a layered learning methodology, memory and monitoring functions are introduced. Finally, we outline a list of future research topics related to the enhancement of such systems.

Keywords

Natural language interfaces, learning control systems, multiple manipulators, cooperative systems, cognitive science

I. Introduction

Endowing a robot system with the ability to carry on a goal-directed multimodal dialogue using natural language (NL), speech, gesture, gaze, etc. for performing non-trivial tasks is a demanding challenge not only from a robotics and a computer science perspective: it cannot be tackled without a deep understanding of linguistics and human psychology [5]. There are two conceptually different approaches to designing an interface architecture for incorporating NL input into a robotic system: the Front-End and the Communicator approach.

A. The Front-End Approach

The robot system receives instructions in NL that completely specify a task the instructor wants to be performed. The input is analysed and the necessary actions are taken in a subsequent separate step. Upon completion of the task, i.e. after having carried out a script invoked by the instruction fully autonomously, the system is ready for accepting new input. This approach is ideal for systems that have to deal only with a limited set and scope of tasks, which do not vary much over time either. Inadvertent changes of the environment resulting from the robot's actions, which would require a re-formulation of the problem, cannot be considered. Neither is it possible to make specific references to objects (and/or their attributes) that are relevant only to certain transient system states because neither the programmer nor the instructor cannot foresee all of these states. Examples for this approach are [6], [1], [10].

To overcome the limitations of this approach, the concept of the "Artificial Communicator" was developed, which we briefly outline in the sequel.

B. The Communicator Approach

If the nature of assembly tasks cannot be fully predicted, it becomes inevitable to decompose them into more elementary actions. Ideally, the actions specified are elementary in such a way that they always refer to only one step in the assembly of objects or aggregates, i.e. they refer to only one object that is to be assembled with another object or collection of aggregates. The entirety of a system that transforms suitable instructions into such actions is called an *Artificial Communicator* (*AC*). It consists of sensor subsystems, NL processing and further cognitive modules and the robotic actors. From the instructor's point of view the AC should resemble a *Human Communicator* (*HC*) as closely as possible [8]. This implies several important properties of AC behaviour:

i) All modules of the AC must contribute to an event-driven *incremental* behaviour: as soon as sufficient NL input information becomes available, the AC must react. Response times must be on the order of human waiting tolerances.

ii) One of the most difficult problems is the disambiguation of instructor's references to objects. This may require the use of sensor measurements such as integration of robot vision or further NL input resulting from an AC request for more detailed information.

iii) In order to make the system's response seem "natural", some rules of *speech act theory* should be observed. The sequence of actions must follow a "principle of least astonishment", i.e. in a given state the AC should take the actions that the instructor would expect it to take. Furthermore, sensor measurements and their abstractions that are to be communicated about must be transformed into a human comprehensible form.

iv) It must be possible for the instructor to communicate with the AC about both scene or object properties (e.g. object position, orientation, type) and about the AC system itself. Examples of the latter are meta-conversations about the configuration of the robot arms or about actions taken by the AC.

v) The instructor must have a view of the same objects in the scene as the AC's perception system.

vi) The AC must exhibit *robust* behaviour, i.e. all system states, even those triggered by contradictory or incomplete sensor readings as well as non-sensical NL input must lead to sensible actions being taken.

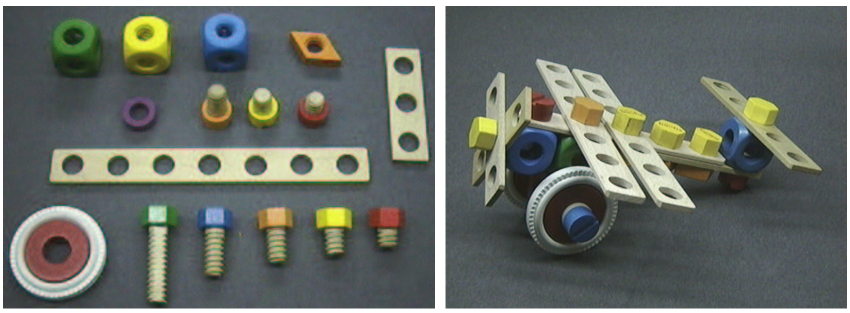
Altogether, the AC must be seamlessly *integrated* into the handling/manipulation process. More importantly, it must be *situated*, which means that the situational context (i.e. the state of the AC and its environment) of a certain NL and input of further modalities is always considered for its interpretation. The process of interpretation, in turn, may depend on the history of utterances up to a certain point in the conversation. It may be helpful, for example, to clearly state the goal of the assembly before proceeding with a description of the elementary actions. There are, however, situations in which such a "stepwise refinement" is counter-productive, e.g. if the final goal cannot be easily described. Studies based on observations of children performing assembly tasks have proven to be useful in developing possible interpretation control flows.

From the engineering perspective, the two approaches can be likened to *open loop control* (Front-End Approach) and *closed loop control* (Communicator Approach) with the human instructor being part of the closed loop. Several projects on communicative agents realised with real robots have been reported, e.g. [9], [2].

Our research work described in the following sections is embedded into a larger interdisciplinary research project aiming at the development of ACs for various purposes and involving scientists from the fields of computer linguistics, cognitive linguistics, computer science and robotics. For performing assembly tasks and to facilitate human interaction with language and gestures, we have been developing a two-arm robotic system to model and realise human sensorimotor skills. This robotic system serves as the major test-bed of the on-going interdisciplinary research program of the project SFB[1] 360 “Situated Artificial Communicators” (SAC) at the University of Bielefeld [13].

II. The Situated Artificial Communicator

There is ample evidence that there exists a strong link between human motor skill and cognitive development, e.g. [7]. Our abilities of emulation, mental modelling and planning of motion are central to human intelligence [3] and, by the way, a precondition for anticipation, but they also critically depend on the experience we make with our own body dynamics as we plastically adapt our body’s shape to the environment. As a basic scenario, the assembly procedure of a toy aircraft (constructed with “Baufix” parts, see Fig. 1) was selected. A number of assembly parts must be recognised, manipulated and built together to construct the model aircraft. In each of these steps, a human communicator instructs the robot, which implies that the interaction between them plays an important role in the whole process (Fig. 2).



(a) The Baufix construction parts. (b) The goal aggregate.

Fig. 1. The assembly of a toy aircraft.

[1] Collaborative research centre funded by the Deutsche Forschungsgemeinschaft (DFG).

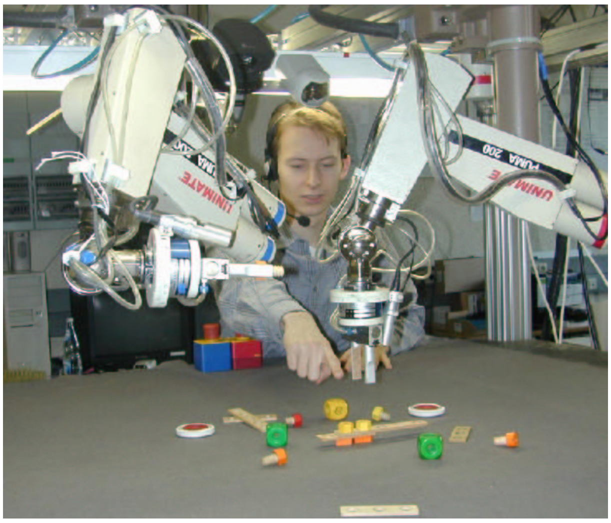


Fig. 2. The two-arm multisensor robot system for dialogue-guided assembly.

The physical set-up of the SAC system consists of the following components:

i) Two six-degree-of-freedom PUMA-260 manipulators are installed overhead in a stationary assembly cell. On each wrist of the manipulator, a pneumatic jaw-gripper with integrated force/torque sensor and "self-viewing" hand-eye system (local sensor) is mounted. As an option, a third manipulator with hand-camera installed on the side can be applied to help with fixating or active exploration tasks.

ii) Two cameras with controllable zoom, auto-focus and auto-exposure provide the main vision function. Their tasks are to build 2D/3D world models, to supervise gross motion of the robot as well as to trace the gesture and gaze of the human instructor.

iii) A microphone and loudspeakers are connected with a standard voice recognition system to transform spoken instructions to word sequences and to synthesize the generated speech output.

III. Control Architecture

As the backbone of an intelligent system, the control architecture of a complex technical system describes the functionality of individual modules and the interplay between them. We developed an interactive, incremental architecture for the SAC according to Fig. 3. A HC is closely involved in the whole assembly process.

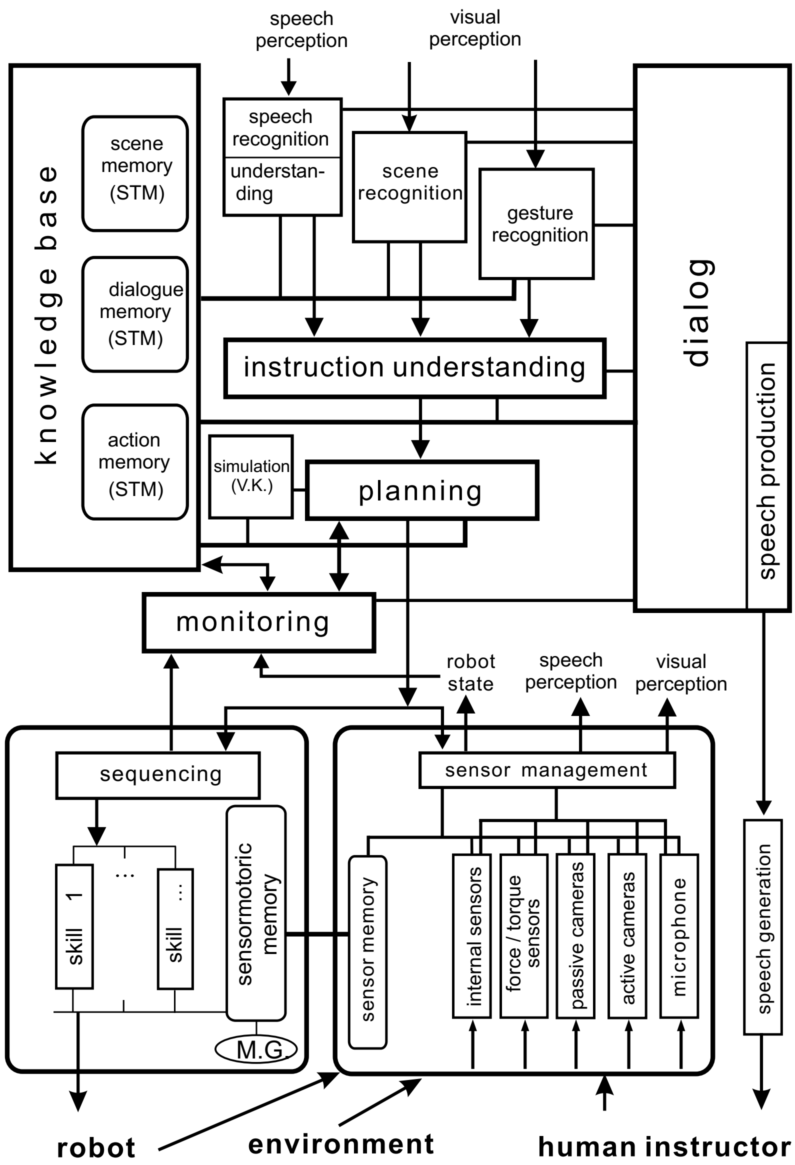


Fig. 3. A control architecture of the Situated Artificial Communicators (SAC) for perception, instruction understanding and execution.

For clarity, the whole architecture is partitioned into three blocks: Perception (right-bottom), High-Level Cognitive Functions (upper half) and Execution (left-bottom).

A. Perception Modules

The tasks of the perception system include self-perception and the perception of the physical environment as well as the human instructor. The complete robot state is specified by the joint and Cartesian positions of the arm, the posture of the hand and the forces/torques exerted on the later. This information can be acquired by the robot internal sensors like encoders/potentiometers and force/torque sensors. The current robot state is the input to the "monitoring" module. Another interesting topic on the supporting role of the robot state to help better understanding emotional intervention instructions like "Haaalt!" when the robot is moving near to the assembly surface. The assembly objects in the environment are observed by multiple cameras – a major function of sensor-based robotics.

To better handle the "human-in-the-loop" problem, human perception is viewed as one important extension of autonomous robots. Therefore, we track visual information about the human instructor like gesture and gaze by the static and articulated cameras. The naturally spoken instructions of the human instructor are input through a microphone and recognised as word sequences. The sensor management contains the data fusion and sensor integration, supplies the specified values of "robot state", "speech perception" and "visual perception". The speech and visual perception results are the main input for the high-level cognitive functions outlined below.

B. High-Level Cognitive Functions

The SAC and the HC interact through natural speech and a small set of hand gestures. First, an instruction is spoken to the robot system and recognised with the speech recognition engine. In the current system, *ViaVoice* recognises only sentences which the grammar we developed allows. In practice, hundreds of grammatical rules can be used. If the recognition succeeds, the results are forwarded to the speech recognition/understanding module.

B.1 Transforming Instructions to Elementary Operations

By their very nature, human instructions are situated, ambiguous, and frequently incomplete. In most cases, however, the semantic analysis of such utterances will result in sensible operations. An example is the command "*Grasp the left screw*". The system has to identify the operation (*grasp*), the object for this operation (*screw*), and the situated specification of the object (*left*). With the help of a hand gesture the operator can further disambiguate the object. The system may then use the geometric knowledge of the world to identify the right object. Other situated examples are: "*Insert in the hole above*", "*Screw the bar on the downside in the same way as on the upside*", "*Put that there*", "*Rotate slightly further to the right*", "*Do it again*", etc.

The output of the analysis is then verified to check if the intended operation can be carried out. If in doubt, the SAC asks for further specifications or it is authorised to pick an object by itself. Once the proper operation is determined, it is given to the execution module on the next level. The final result on this level consists of an *Elementary Operation* (EO) and the objects to be manipulated with the manipulation-relevant information such as type, position/orientation, colour, pose (standing, lying, etc).

An EO is defined in this system as an operation which does not need any further action planning. Typical EOs are: *grasp*, *place*, *insert into*, *put on*, *screw*, *regrasp*, *alignment*. The robustness of these operations mainly depend on the quality of the different skills.

B.2 Planning and Monitoring

Based on the planning module, an assembly task of the toy aircraft, or of sub-aggregates, is decomposed into a sequence of EOs. The final decision about the motion sequence depends on the instructions of the human user as well as the generated plan. The *planning* module should not only be able to understand the human instructions, but it should also learn from the human guidance and improve its planning abilities gradually. It receives an EO from the *instruction understanding*. By referencing the *action memory*, the *planning* module chooses the corresponding basic primitive sequence for the operation. This sequence is a script of basic primitives for implementing the given EO. The task here includes planning of the necessary trajectories, choosing the right robot(s) and basic exception handling.

Monitoring plays an important role to make an intelligent system robust. It is also used frequently by a human-being in manipulation and speaking, especially in a new environment or for a new task. Monitoring and potential re-planning for repair actions result in the non-linearity of the understanding-planning-execution cycle, but they represent one essential function in the cognitive architecture of a robot. Furthermore, it is meaningful to add a diagnosis function which can provide hypotheses about the reasons of diverse failures.

The unexpected events during the robot action can be for example: *A force exceeds a defined threshold; a camera detects no object; singularities; collisions;* etc. If such an event occurs, it is reported to the planning module.

The *planning* module receives an event report that is generated by the *execution* module described below. In normal operations, the *monitoring* module updates the action memory. It also detects the event failures. If it is found that the robot can continue and/or take repair actions, the *planning* module will generate an appropriate plan. Otherwise, the *monitoring* module sends a request to the *dialogue* module to ask the human communicator how to handle the exception and waits for an instruction. After the execution of each operation, the *knowledge base* is updated.

B.3 Memories

In the knowledge base, only semantic and procedural knowledge are used. In our current implementation this knowledge is still hard-coded. It can be viewed as long-term-memory to a certain degree, which will be extended by learning approaches in our future research. Short-term memories exist in perception modules, which are used for scene recognition, dialogue preparation and action (sensorimotor functions). Learning of another important type of memories, the episodic memory, has been preliminarily studied for the assembly scenarios.

According to empirical investigations, the episodic memory represents one of the most important components of human intelligence. Reminding, mental simulation and planning use episodic memory as the basis. The diverse multisensor data with high bandwidth of our robot such as vision system, joint angles, positions, force profiles etc., can obviously not be saved in their raw format for an arbitrarily long period of time. Therefore, coding approaches based on appearances and features are suggested for summarising and generalising experiences from the successfully performed operations. The multisensor trajectories and the motor signals are "grounded" in the learned operation sequences. Fig. 4 depicts the instruction for building an "elevator control" and the corresponding sensor trajectory.

C. Execution Functions

Sequences are executed by the *sequencer*, which activates different skills on the *execution* level.

C.1 Robot Skill Library

The complexity of the skills can range from opening the hand to collision-free control of the two arms to a meeting point. Advanced skills are composed of one or more basic skills. Generally, three different kinds of skills are defined :

i) Motor skills: *Open and close gripper; Drive joint to; Drive arm to; Rotate gripper; Move arm in approach direction; Move camera, etc.*

ii) Sensor skills: *Get joint; Get position in world; Get force in approach direction; Get torques;*

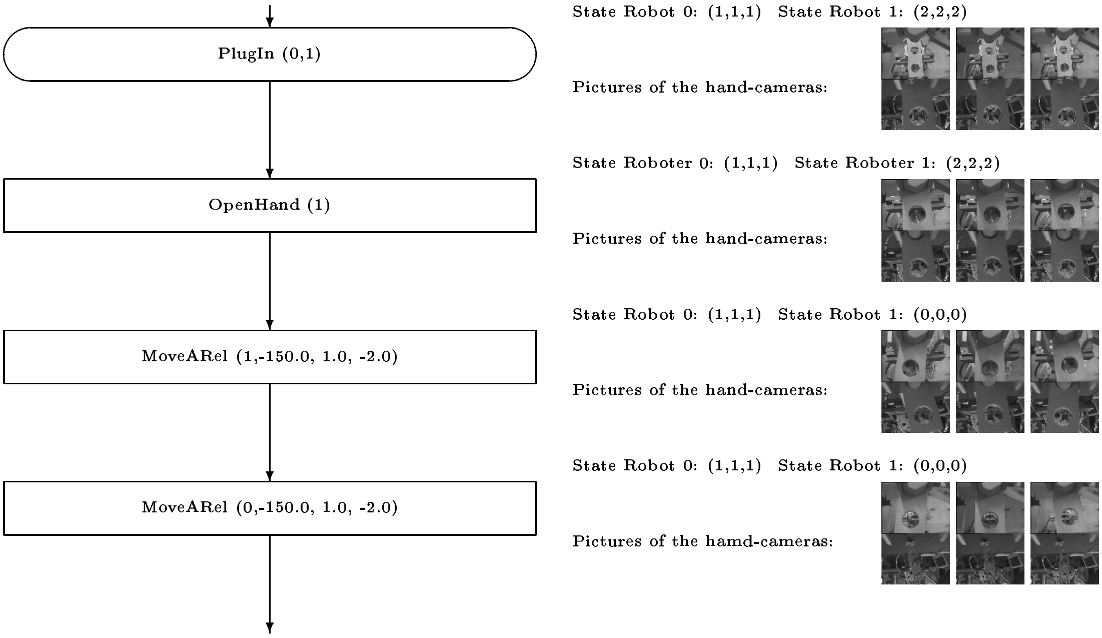


Fig. 4. An instruction sequence with sensor trajectory for building an "elevator control". The parameters of the instructions "PlugIn", "OpenHand" and "MoveARel" (relative movement along the Approach-axis of the tool frame) can be flexibly set. Different instruction sequences leading to the same aggregate are fused by a generic approach.

Check if a specific position is reachable; Take a camera picture; Detect object; Detect moving robot; Track an object, etc.

iii) Sensorimotor skills: *Force-guarded motion; Vision-guided gross movement to a goal position; Visual servoing of the gripper to optimal grasping position, etc.*

C.2 Control by a Neuro-Fuzzy Model

We developed a universal neuro-fuzzy method as the underlying model for robot skill learning [12]. Our experimental results show under the most diverse conditions that we can extract geometric features based on the calculations of moments to encode the positioning information and to find non-geometric parameters based on combining principal components. Therefore, if the input is high-dimensional, an efficient dimension reduction can be achieved by projecting the original input space into a minimal subspace.

Variables in the subspace can be partitioned by covering them with linguistic terms (the right part of Fig. 5). In the following implementations fuzzy controllers constructed according to the B-spline model are used [11]. This model can be classified as an adaptive, universal function approximator using regularisation approaches. It provides an ideal implementation of the CMAC model (cerebellar model articulation controller).

We define linguistic terms for input variables with B-spline basis functions and for output variables with singletons. This method requires fewer parameters than other set functions such as trapezoid, Gaussian function, etc. The output computation is very simple and the interpolation process is transparent. We also achieved good approximation capabilities and rapid convergence of the B-spline controllers. Both self-supervised and reinforcement learning have been applied to this model to realise most of the sensorimotor skills [12].

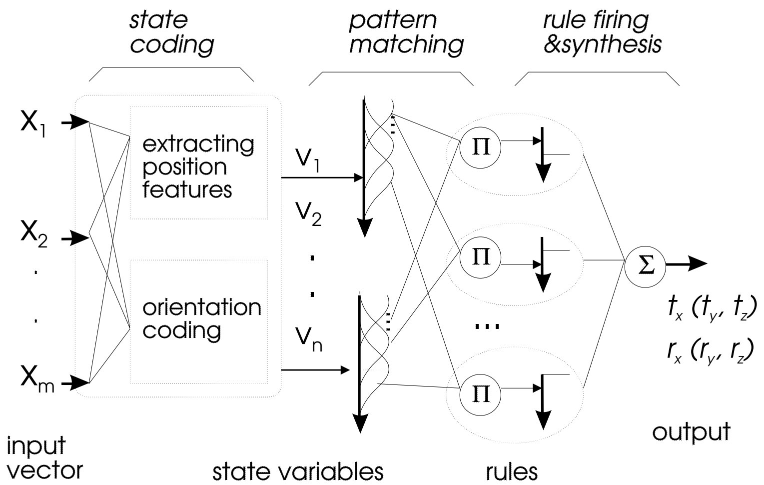


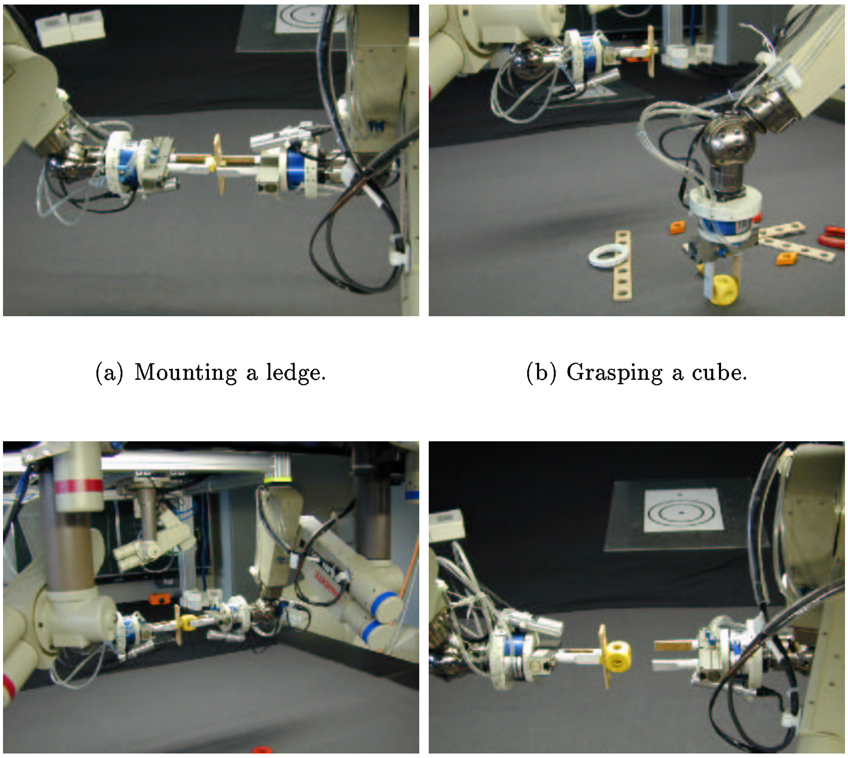
Fig. 5. The perception-action mapping is realised based on a neuro-fuzzy model.

D. Layered Learning

Learning the interplay of perception, positioning and manipulation is the foundation of a smooth execution of a command sequence of a human instructor. If a command refers to an EO, the disambiguation of the instruction based on multimodality is the key process. The autonomous sensor-based execution of these instructions requires adaptive, multisensor based skills with an understanding of a certain amount of linguistic labels. If complex instructions are used, however, the robot system should possess capabilities of skill fusion, sequence generation and planning. It is expected to generate the same result after a repeated instruction even if the starting situation has changed. The layered learning approach is the scheme to meet this challenge.

Under this concept, tasks are decomposed from high to low level. Real situated sensor and actuator signals are located on the lowest level. Through task-oriented learning, the linguistic terms for describing the perceived situations as well as robot motions are generated. Skills for manipulation and assembly are acquired by learning on this level using the above mentioned neuro-fuzzy model. Furthermore, the learning results on the lower levels serve as the basis of the higher levels such as EOs, sequences, strategies, planning and further cognitive capabilities.

To learn the operation sequences automatically for two-arms, we developed a method for learning cooperative tasks. If a single robot is unable to grasp an object in a certain orientation, for example, it can only continue with the help of other robots. The grasping can be realised by a sequence of cooperative operations that re-orient the object. Several sequences are needed to handle the different situations in which an object is not graspable for the robot. It is shown that a distributed learning method based on a Markov decision process can learn the sequences for the involved robots, a master robot that needs to grasp and a helping robot that supports it with the re-orientation. A novel state-action graph scheme is used to store the reinforcement values of the learning process [4]. Fig. 6 shows an assembly process learned by the state-action graph representation. The aggregate composed of a screw, a ledge and a cube is to be assembled. We use the object description and its graph matching algorithms to find out whether the object to construct is a sub-assembly of the goal aggregate (positive reward) or not (negative reward). This will give a reward whenever a part is successfully attached to the growing aggregate.



(a) Mounting a ledge.

(b) Grasping a cube.

(c) Mounting the cube.

(d) Goal state reached.

Fig. 6. The learned assembly process for building a simple aggregate.

E. Experimental Results

Fig. 7 shows typical aggregates that can be built with the set-up as developed up to now. Here we briefly describe a sample dialogue which was carried out between the SAC and the HC in order to build the “elevator control” aggregate (no. 1 in Fig. 7) of the aircraft out of three elementary objects. The objects were laid out on the table, and there were many other objects positioned in an arbitrary order on the table than necessary. The HC had a complete image in mind of what the assembly sequence should be. Alternatively, one could have used the assembly drawings in the construction kit’s instructions and translated them into NL.

The first SAC input request is output after it checked that all modules of the setup are working properly. The necessary classification and subsequent steps are based on the colour image obtained from the overhead colour camera. After the SAC finding out if all objects are present and after going through an optional object naming procedure, the HC input “Take a screw!” first triggers the action planner, which decides which object to grasp and which robot to use. Since the HC did not specify either of these parameters, both are selected according to the principle of economy. In this case, they are so chosen as to minimise robot motion. The motion planner then computes a trajectory, which is passed to the RCCL/RCI subsystem (*Robot Control C Library/Real-time Control Interface*). Since there are enough bolts available, the SAC issues its standard request for input once the bolt is picked up.

An HC input “Now, take the three-hole slat!” results in the other robot picking up the slat. Before this may happen, however, it has to be cleared up, which slat to take (SAC: “I see more than one such slats” and HC: “Take this one!” <points to one>). This involves the incorporation of the gesture recogniser. Under the instruction “Screw the bolt through the slat”, the screwing is triggered, involving the peg-in-hole EO mentioned above followed by the screwing EO. For reasons of space the subsequent steps of the dialogue have to omitted here; they show how error handling and many other operations can be performed – most of which humans are not aware of when they expect machines to do “what I mean”.

IV. Future Work

Among many topics to be explored in the future research, some important ones can be listed as follows:

Seamless communicator. Interfaces will be closely coupled with planning and monitoring. Ideal action needs to be inferred based on motion and action planning while considering the context and the human preference.

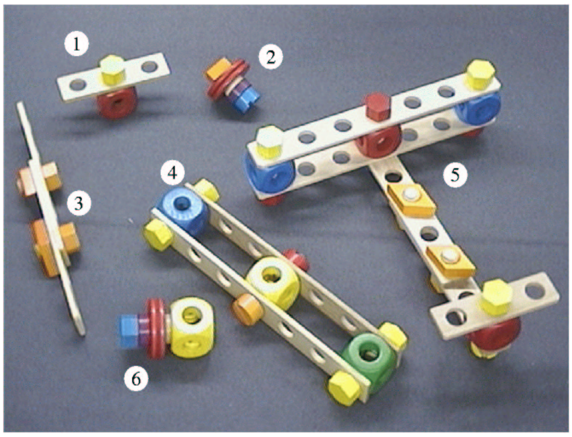


Fig. 7. Finished aggregates that can currently be built in multimodal dialogues by the SAC assembly robot.

Active intention detection based on multiple cues. Speech, gesture, motion sequences (human demonstrations) will be integrated and combined with contexts, knowledges and personal preference. The cross-modal interplay will be investigated. Since the system resources are limited, sensory input needs to be selected by using factor analysis, signal synthesis and tracking focus of interests.

General human perception. Human motions are captured without using artificial markers. Wide-range, active camera configurations are applied to human recognition and precise gaze perception, also by low-quality input and occlusions. The robustness of the voice input in real environments should be significantly improved. This task is even more challenging if non close-speaking microphones are used.

Grounded learning of multisensor events, sequences and human activities. The long-term-memory should be learned from the short-term-memory so that symbols, sequences, names and attributes are anchored in the real sensor/actuator world. To enable the arbitrary transition between digital measurements and concepts, symbolic sparse coding, granular computing, fuzzy sets and rough sets will be investigated and integrated. The sensor capability can be extended by using linguistic modelling of human perception and sensor fusion so that information which is difficult to measure, incomplete or noisy can be perceived. Learning on the higher level should be conducted to select action strategies and to generate intelligent dialogues. This will need the tight integration of more components and more knowledge. The combination of grounded learning and communication will make the human-robot interaction work like interaction with a growing child which will be really entertaining.

Acknowledgement

This research is supported under grant SFB 360 by DFG, the German Research Council.

References

[1] R. Bischoff and V. Graefe. Integrating vision, touch and natural language in the control of a situation-oriented behavior-based humanoid robot. In *IEEE International Conference on Systems, Man, Cybernetics, Tokyo*, 1999.

[2] R. A. Brooks, C. Breazeal, M. Marjanovic, and B. Scassellati. The Cog project: Building a humanoid robot. In C. L. Nehaniv, editor, *Computation for Metaphores, Analogy and Agents*, volume 1562 of *Lecture Notes in Computer Science*, pages 52–87. Springer, 1999.

[3] A. Clark and R. Grush. Towards a cognitive robotics. *Adaptive Behavior*, 7(1):5 – 16, 1999.

[4] M. Ferch and J. Zhang. Learning cooperative grasping with the graph representation of a state-action space. *Journal of Robotics and Autonomous Systems*, 38(3-4) 183 –196, 2002.

[5] C. Grangle and P. Suppes. *Language and Learning for Robots.* CSLI Publications, Stanford, Ca., 1994.

[6] Th. Laengle, T. C. Lueth, E. Stopp, and G. Herzog. Natural language access to intelligent assembly robots: Explaining automatic error recovery. *Artificial Intelligence: Methodology, Systems, Applications*, 1996.

[7] G. Lakoff. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind.* University of Chicago Press, 1990.

[8] R. Moratz, H. Eikmeyer, B. Hildebrandt, A. Knoll, F. Kummert, G. Rickheit, and G. Sagerer. Selective visual perception driven by cues from speech processing. In *Proc. EPIA 95, Workshop on Appl. of AI to Rob. and Vision Syst.*, TransTech Publications, 1995.

[9] K. R. Thorissen. *Communicative Humanoids - A Computational Model of Psychosocial Dialogue Skills.* PhD thesis, MIT Media Lab., 1997.

[10] J. Weng, C. H. Evans, W. S. Hwang, and Y.-B. Lee. The developemental approach to artificial intelligence: Concepts, developmental algorithms and experimental results. In *In Proc. NSF Design & Manufacturing Grantees Conference*, 1999.

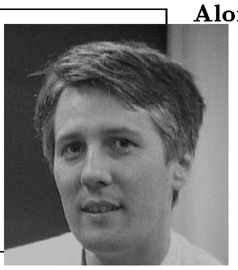
[11] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems*, 13(2/3):257–285, Februar/March 1998.

[12] J. Zhang and A. Knoll. *A Neuro-Fuzzy Learning Approach to Visually Guided 3D Positioning and Pose Control of Robot Arms.* In "Biologically Inspired Robot Behavior Engineering", edited by R. Duro, J. Santos and M. Grana, Springer Verlag, 2001.

[13] J. Zhang, Y. von Collani, and A. Knoll. Interactive assembly by a two-arm robot agent. *Journal of Robotics and Autonomous Systems*, 29:91–100, 1999.

Jianwei Zhang

Dr. Zhang Jianwei received the Bachelor of Engineering (1986, with distinction) and Master of Engineering (1989), both from the Department of Computer Science of Tsinghua University, Beijing, China, and the PhD (1994) at IPR (Institute of Real-Time Computer Systems and Robotics), Department of Computer Science, University of Karlsruhe, Germany. From August 1994 to July 2002, he was Assistant Professor in the Department of Technology, University of Bielefeld, and since August 2002 he is full-professor and the director of the group "Technical Aspects of Multimodal Systems", Department of Computer Science, University of Hamburg, Germany. His research interests are robot control architecture, planning, sensor-based manipulation, robot learning, man-machine-interface and application of neuro-fuzzy systems. In these areas he has published over 90 journal and conference papers, technical reports, five book chapters and two research monographs. He leads several part projects in the Collaborative Research Centre "Situated Artificial Communicators" and coordinates the European Interest Group on "Skill Learning / Multimodal Interaction". He also leads several application projects in laboratory service robots, mobile robots and sensing devices. He is member of the IEEE and GI.



Alois Knoll

Alois C. Knoll received the diploma (M.Sc.) degree in Electrical Engineering from the University of Stuttgart, Germany, in 1985 and his PhD (summa cum laude) in computer science from the Technical University of Berlin, Germany, in 1988. He served on the faculty of the computer science department of TU Berlin until 1993, when he qualified for teaching computer science at a university (habilitation). He then joined the Technical Faculty of the University of Bielefeld, where was a full professor and the director of the research group Technical Informatics until 2001. In March 2001 he was appointed to the board of directors of the Fraunhofer-Institute for Autonomous Intelligent Systems. Since autumn 2001 he has been a professor of computer science at the Computer Science Department of the Technical University of Munich. His research interests include sensor-based robotics, multi-agent systems, data fusion, adaptive systems and multimedia information retrieval. In these fields he has published over eighty technical papers and guest-edited international journals. He has been part of (and has coordinated) several large scale national collaborative research projects (funded by the EU, the DFG, the DAAD, the state of North-Rhine-Westphalia). He initiated and was the program chairman of the First IEEE/RAS Conference on Humanoid Robots (IEEE-RAS/RSJ Humanoids2000), which took place at the Mass. Inst. of Technology (MIT) in September 2000. Prof. Knoll is a member of the German Society for Computer Science (Gesellschaft fuer Informatik (GI)) and the IEEE.