Paper 21

A Basic System for Multimodal Robot Instruction

Alois Knoll and Ingo Glöckner
Technische Fakultät, Universität Bielefeld
{knoll|ingo}@techfak.uni-bielefeld.de

1 Introduction

Due to recent developments in enabling technologies Brooks and Stein (1994) (processing power, mechatronics, walking machines, articulated vision heads and more) but also due to findings and developments in other fields (e.g. studies of the human brain, linguistics, psychology), we currently observe a shift in the view of what artificial intelligence is and how it can be put to work in operational autonomous systems. This sets the stage for putting perceptive, cognitive, communicative and manipulatory abilities together to create truly interactive robot systems.

In the past, there have been a number of attempts to teach robots by showing them a task to be performed. We note, however, that such systems for "teaching by demonstration" or skill transfer have not met with much success. We identify three main reasons for this failure: **(i)** Instruction input is monomodal, mostly through a fixed camera. This precludes the system from constructing cross-modal associations by evaluating clues from more than one modality. It also prevents the instructor from giving additional explanations in "natural" modalities, e.g. teaching movements of the hand supplemented by instructive speech statements. **(ii)** Partly due to monomodality the instruction is not in the form of a dialogue between the instructor and the robot. Dialogue-oriented interaction may be the source of additional information in "normal" instruction mode, but it becomes indispensable in the case of error conditions.

2 Human-Humanoid Interaction

In view of the aforementioned needs and deficiencies we present some of our theoretical work involving methodology form linguistics and robotics. We intend to show how future robot systems will be able to carry on dialogues in several modalities over selected domains. Endowing a humanoid robot with the ability to carry a goal-directed multimodal dialogue (vision, natural language (NL), speech, gesture, face expressions, force, ...) 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 deeper understanding of linguistics and human psychology Grangle and Suppes (1994). There are two conceptually different approaches to designing an architecture for incorporating NL input into a robotic system: the Front-End and the Communicator approach.

The "Front-End" Approach. The robot system receives instructions in NL that completely specify a – possibly very complex – task the instructor wants to be performed. Examples are Restaino and Meinicoff (1985); Kawamura and Iskarous (1994); Laengle et al. (1995). The input is analysed and in a subsequent separate step the necessary actions are taken. Upon completion of the task, i.e. after having carried out a script invoked by the instruction fully autonomously, the system is

BI-DIALOG 2001 (P. Kühnlein, H. Rieser, & H. Zeevat (eds)).

Technische Fakultät, Universität Bielefeld
{knoll|ingo}@techfak.uni-bielefeld.de.

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. It much less lends itself to tasks that presuppose a high degree of flexibility during their processing. Inadvertent changes of the environment resulting from the robot's actions, which would require a re-formulation of the problem, cannot be considered. Such situations cannot be dealt with unless the whole decisionmaking competence is transferred to the robotic system. For non-trivial tasks this is currently impossible; it is questionable whether it is at all desirable to try not to make use of the instructor's sensory system and intelligence (see the discussion of rationales for the introduction of sensor-based manipulation primitives in Hirzinger et al. (1994)). Neither is it possible to make specific references to objects (and/or their attributes) that are relevant only to certain transient system states because the instructor cannot foresee all of these states (cf. the well-known AI "frame problem"). These references, however, are often indispensable for the system to work correctly, i.e. as intended by the instructor. With this approach the system cannot produce requests for specific and more detailed instructions because those, too, may arise only during the sequence of actions.

Communicator or Incremental Approach. If the nature of tasks cannot be fully predicted, it becomes inevitable to decompose them into (a set of) more elementary actions. Ideally, the actions specified are atomic 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 thereof (aggregates). The entirety of a system that transforms suitable instructions into such actions is called an *artificial communicator* (*AC*). It consists of cognitive NL processing, sensor subsystem and the robotic actors. From the instructor's point of view the AC should resemble a human communicator (*HC*) as closely as possible Moratz et al. (1995). 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 reaction delays. **(ii)** One of the most difficult problems is the disambiguation of instructor's references to objects. This may require the use of sensor measurements or 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 speechact theory should be observed. The sequence of actions must follow a "principle of least astonishment", i.e. 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 (optical) sensors. **(vi)** The AC must exhibit *robust* behaviour, i.e. all system states, even those triggered by contradictory or incomplete sensor readings as well as nonsensical NL input must lead to sensible actions being taken.

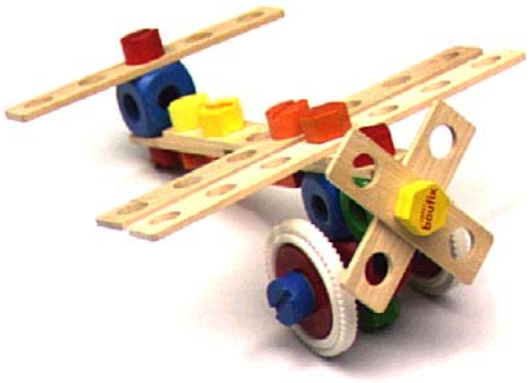

Figure 21.1: The fully assembled "aircraft".

In other words: 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 further modalities) input 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 atomic actions. There are, however, situations in which such a "stepwise refinement" is counterproductive, 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 an engineering perspective the two approaches can be likened to *open loop control* (Front-End Approach) and *closed loop control* (Incremental Approach) with the human instructor being part of the closed loop.

3 Scenario for Practical Evaluation

For studying situated goal-directed multimodal assembly dialogues, a prototypical scenario was chosen carefully. In this scenario a human instructor and an AC cooperate in building aggregates from elements of a toy construction set intended for children of the age of 4 years and up. The elements are made of wood (with little precision); their size is well suited to the parallel jaw grippers of our robots. The goal pursued in the sample conversations is the construction of the "aircraft" shown in fig. 21.1.

Due to several mechanical constraints its complete construction is difficult for children. As observed during some of the experiments even some adults had problems assembling the aircraft although they were provided with the exploded view of the assembly. It remains to be shown that this can be done with robots using no specialised tools. In principle, however, it may one day become possible to replace the HC with an AC and to achieve the same goals through the same dialogue.

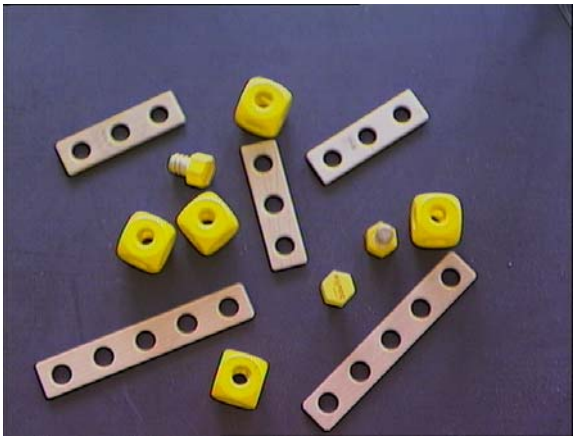

Figure 21.2: Randomly positioned construction elements: Cubes, Slats, Bolts.

To illustrate *only one individual problem* occuring from a linguistic point of view, we briefly turn to the question of *object naming* in this scenario. In an assembly dialogue between HCs each object of the scenario may be referenced using a variety of different names. Before a sensible dialogue between HC and AC may take place, however, an unambiguous binding between an object and its reference name must be established. This binding must be identical on both the HC and AC side. Since there is no common naming convention in natural language that is precise enough, a straightforward way of generating (initial) bindings is negotiation. Before entering the assembly, object names are assigned in an opening phase. The AC might, for example, point at one of the objects of fig. 21.2 (e.g. by highlighting it on a monitor) and ask the HC "What do we call / What do you want to call this object?" The HC's answer is then used as the name for the remainder of the assembly session.

While acceptable for testing purposes, such a procedure is obviously too inconvenient, time consuming and hence impractical in real-world applications involving dozens of objects. This is the reason, therefore, that the AC must possess the ability to react in a flexible manner to all (most) of the conceivable object names. It would be both difficult, cumbersome and intractable in the general case to compile all possible names for all possible objects in all possible situations. Fortunately, linguistic experiments have shown that rules may be postulated that HCs obey in assembly-type dialogues. These rules can be used to reduce the "name space" the AC must consider. Some of them follow: **(i)** Even with simple items like the cube in fig. 21.2, HCs frequently switch between names. Apart from *cube* the object ist called *die, dice or block*. **(ii)** An object may be referenced not by its generic name but by its function in the situational context: the slat is named as such but also as *wing*, the cube may be called *nut* when used as the counterpart of the bolt. **(iii)** Particularly in this scenario objects are named after their geometrical shape where frequently a projection from three into two dimensions can be observed, e.g. the cube becomes a *square*.

The AC must recognise and cope with the principles and conditions under which these transformations occur Heydrich and Rieser (1995).

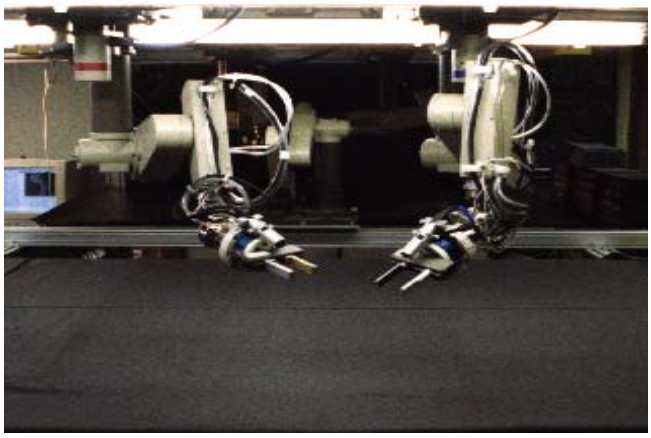

Figure 21.3: A view of the set-up for assembly.

4 Dialogue Control in Action

Even the construction of an aggregate of only a few elements may consist of a great number of elementary actions. Every assembly step resulting from an instruction comprises three distinct phases:

- The recording of the scene content using the sensory system;
- the processing of what is seen/sensed and the development of a plan for achieving a set (sub-)goal;
- the assembly of available elements with the actors.

In other words: Every assembly step is composed of perceptive, cognitive and manipulative actions. Each of these may be atomic or complex and mirror (i.e. is the consequence of) specific instructions given by the HC.

While system architectures are conceivable that implement a temporally interleaved processing of perception, cognition and action, our system currently works strictly sequentially. At the beginning of each individual assembly step the scene is analysed visually. The objects are detected and their locations are computed. A geometrical model of the scene is generated. Once this model is available, the AC requests an instruction from the HC (the instructor). These instructions can be of the type

- Assembly (Construction): "Take the red screw";

- Scene Control: "The screw is/should be located on the left hand side of the bar";
- Meta-Level-Control: "Move the elbow up a little" or "Turn this camera a little clockwise";

where the latter type of meta-level instructions very rarely occurs in human construction dialogues. The instructions are analysed linguistically and interpreted according to a hypotheses model of the scene and the history of the construction process, e.g. taking into account that a robot that has already grasped an object cannot grasp another one. As part of the cognitive phase a simple planner transforms complex into atomic actions.

Unlike standard motion sequence planners, this planner must also draw on knowledge obtained from cognitive linguistic observations. For example, an HC does not necessarily give all the instructions required for fulfilling the preconditions of a certain manipulative action. In some sense the problem is underdetermined; the planner must provide a solution within the given degrees of freedom. A simple example: The HC would not instruct the AC to grasp a screw (let alone a specific screw if more than one is available), before giving an instruction involving a screw. The reasoning about what the HC may have meant and the necessary inferences are left to the AC's planner with no help other than the cognitive knowledge mentioned above. Currently, in such a situation, our system selects the object that the robot can grasp most easily (following a *principle of economy*). In the future this will be extended in such a way as to make an attention control possible, i.e those objects are chosen that are in the *focus of the discourse*.

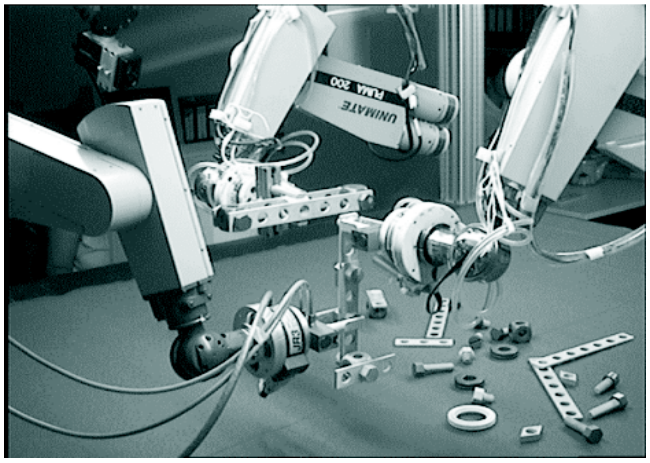

Figure 21.4: A view of the flexible assembly cell in action.

Meta-level instructions/statements are necessary for interrupting the dialogue whenever the HC wants to guide the AC to a better sequence of actions than the latter is able to find autonomously. This is in contrast to most meta-level utterances in human dialogues, which normally deal with the (format of) the dialogue itself ("What are you doing there?", "Be more polite!").

Another important application of these instructions is error handling: imagine a situation in which the robot arm has run into a singularity while following move instructions by the HC. The typical HC, of course, has no comprehension of this problem. In such a case the AC must explain the (imminent) error, and a dialogue must be conducted about possible (consensual) ways leading out of the error situation. Sometimes errors pertaining to the actuators may be anticipated. If in such a case proper use is made of the NL-production facility of the AC, errors may even be *prevented*. A further source of errors are utterances by the HC that the AC does not understand correctly. If the AC fails to comprehend the meaning of a statement, the HC must recognise the AC's problem and act accordingly. For this reason the linguistic components were so designed as to provide transparent messages whenever an error occurs. There are three classes of errors: lexical, syntactical and semantical. The reason for a lexical error is a certain (uncommon) word missing in the system's lexicon or a word having been misspelled. A syntactic error is reported when the parser cannot combine the individual words, i.e. it

cannot compile a sensible syntactical structure. A semantic error occurs if the action required by the HC cannot be taken. This normally happens when the preconditions of the action are not met (and the necessary steps cannot be inferred); in particular if the necessary objects are not present in the scene. After completion of the perception-cognition-manipulation sequence for a single assembly step, this cycle is repeated until the aggregate is finished.

4.1 Experimental Setup

To complement the AC's cognitive component a manipulation unit or *cell* was built using standard robots that cooperate and come as close as possible to the geometry of humans and their hand/arm. The similarity of the geometry often makes it easier for an HC giving instructions to the AC to image himself into the problems arising from the AC's point of view. It also enables the immediate transfer to a humanoid (torso). The cell mimics the situation of an assembly carried out by an HC sitting at a table (and possibly being given instructions). In such a setting the construction elements are placed on the table, the HC's arms/hands cooperate from above the table.

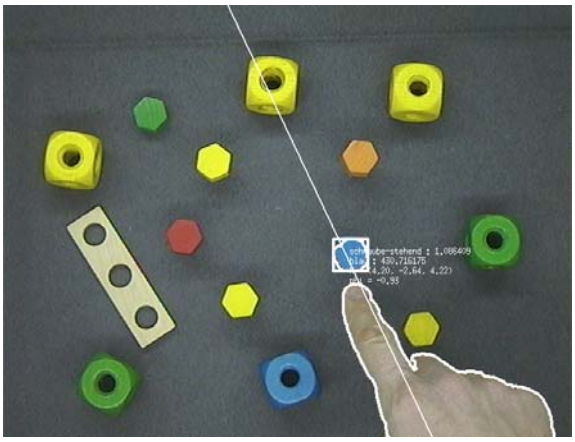

Figure 21.5: Recognition of simple gestures for identifying NL reference to a certain object

Up to now the following assembly skills have been implemented on both manipulators:

Pick-up: Most elements of the construction set can be picked up from any location on the table. The approach of the end effector's tip to the desired grasping point is controlled in real time using "self-viewing visual servoing" Meinicke and Zhang (1996).

Put-down: Elements or aggregates can be put on the table or on other objects. Prior to releasing the gripper controlled forces and torques may be applied to the object.

Peg-in-hole: Most combinations of objects that can be passed through one another can be handled. If necessary, a reflex can be activated that lets one of the robots find the center of the hole by following a spiral path under force control.

Screwing: This is by far the most complex operation available. It requires sensitive force/motion control. It involves the **(i)** approach phase in which the true thread position is determined; detection of the contact angle between screw and start of the thread; **(ii)** re-grasping of the bolt head after completing one revolution; **(iii)** application of the tightening torque. The latter is particularly difficult because the wooden screws tend to block. Special types of adaptive fuzzy controllers for force control have proven to be superior in performance to standard PID controllers Zhang et al. (1997).

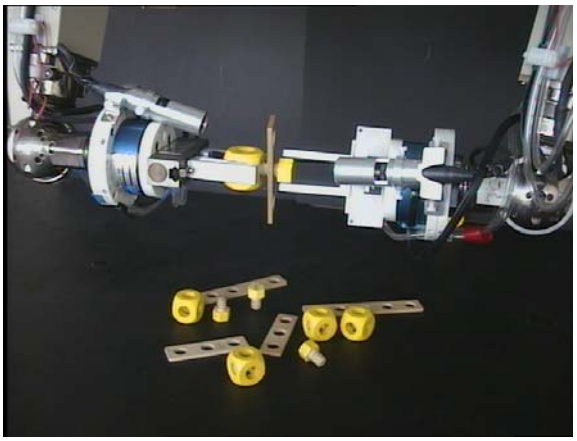

Figure 21.6: Screwing by cooperating robots.

4.2 Sample Dialogue and Results

Table 21.1 shows the beginning of a sample dialogue which was carried out in order to build the "elevator control" aggregate of the aircraft (fig. 21.1) out of three elementary objects. The objects were laid out on the table in a way similar to fig. 21.2 (i.e. there were many more objects positioned in arbitrary order on the table than necessary). The instructor had a complete image in his mind of what the assembly sequence should be. Alternatively, he could have used the assembly drawings in the construction kit's instructions and translated them into NL.

Initialisation			
(2) Constructor: *Yes, let's get started* Confirmation Initialisation	(i) No, not today (ii) One moment, please! [need to have my hands free] ...	Initialise hardware/software	
(3) **Today, we want to build a [Baufix-] aircraft [together] [, we'll start with the elevator control]!** Problem Specification		Activate domain knowledge	– Only sensible, if knowledge about object domain has been acquired – "Baufix" as opposed to "Lockheed" specifies domain (properties) – "Build..." focuses on target object, "Build together..." focuses on cooperation
(4) *All right!* Confirmation problem specification	(i) I know nothing about these aircrafts (ii) Again!	(Enter learning mode)	Teaching: "An aircraft is ..." or: Discussion about assembly plan
(5) *I'll check whether there are all necessary objects available.* Preconditions of action	I need another screw [, cube ,...].	Object recognition in scene Update scene model Problem formulation output	Precondition: Common field of view Ins/Cons Cons has knowledge about necessary objects
(6) *What would you call the [rectangular] object [in the upper right corner, to your left hand side, to my left, ...]?* Negotiation of object naming (conventions)		Learning/update naming in knowledge base (baptising act)	Precondition: – Common field of view Ins/Cons – Negotiations about naming and locations only sensible if an abstract object model is available to Cons
(7) **This is a cube!** <**points to it**> Object naming		Focus to hand Recognition of gesture	
(8) *OK, that's what we'll call it!* Accept bject naming			Gender and further properties can only be determined if there is an entry in the computer lexicon
(9) **Take a screw!** Instruction	[First,] you need a screw!	Find reference object in scene	– Object indefinite, Cons selects on its own – Alternative is an indirect instruction that needs not be exectued immediately [but before any others]
(10) *I am taking one!* Commenting action	(i) ...+ with my left hand (ii) I canot see a screw	Motor and Sensorimotor control	(i) Cons expresses selection (ii) Cons signals that it knows about the importance of the hand and indicates its orientation
(11) **Now, take the three-hole slat!** Instruction	...+ with three holes!	Infer that we need another arm	Definite object naming, works only if there is only one slat in the scene.
(12) *I see more than one [such] slats.* Cons' Identification of Contradictions or Ambiguities	(i) ...+ I would [rather] take the one on top. (ii) ...+ which {one | of the ones I see } do you want me to take?	Detect ambiguities	(i) Cons makes full use of its autonomy (ii) Cons produces two utterances: problem statement and request for information (object spec) it needs. How precise must it be?
(13) **Take this one!** <**points to one**> Instructor's resolution of contradictions	(i) Take the one I am pointing to! (ii) Take the one to {my, your} left! (iii) Take the one you want <and/or suitable gesture> (iv) Take the right one		(i) Makes sure Cons/Ins refer to the same object (ii) Needs reference frame (and info about Ins' location) (iii) E.g. nodding (iv) Location instead of colour/shape
(14) *I have got it.* Action Confirmation	And now the screwing?		Anticipation of the most probable follow-up action
(15) **Screw the bolt through the slat!** (Erroneous) Instruction	(i) Insert the screw {into | through} the slat! (ii) Put the slat on the screw! (iii) Insert the screw through the center hole!	Inference processes about the functions of the objects	Roles and object functionality (do not) match (i) Syntactic structure matches the roles of the objects (ii) Correction of the roles (iii) Instruction to avoid Cons' info request
...	...	...	...

Table 21.1: An excerpt form a sample dialogue, as partly implemented on the set-up.

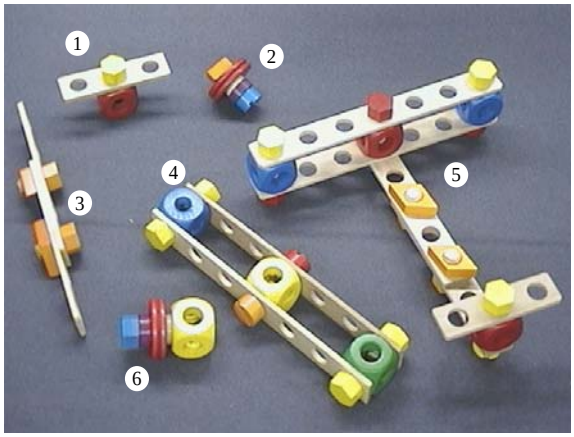

Figure 21.7: Finished aggregates that can currently be built in multimodal dialogues.

Lines input by the HC are typeset in **bold face**; AC output is in *italics*. The first AC input request in *Line 1* is output after it checked 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 AC finding out if all objects are present and after going through an optional object naming procedure (*Lines 6...8*) the HC input in *Line 9* 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 subsystem. Since there are enough bolts available, the AC issues its standard request for input once the bolt is picked up.

HC input *Line 11* results in the other robot picking up the slat. Before this may happen, however, it has to be cleared up, which slat to take (*Lines 12...14*). This involves the incorporation of the gesture recogniser (fig. 21.5). In *Line 15* the screwing is triggered, involving the peg-in-hole module mentioned above followed by the screwing module. The screwing is shown in fig. 21.6. Many uncertain parameters have to be taken into account; in particular the bolt axis is never in line with the effector's z-axis. Using the adaptive force control mentioned above, however, angles between the two axes of up to 15 degrees can be accomodated without blocking (if the thread of the bolt is not excessively worn out). 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 do do "what I mean". Fig. 21.7 shows typical objaect that can – in principle – be built with the setup as developed up to now.

5 Conclusions

We introduced a scenario and a robot system that experimentally show the way humans may communicate with robot systems (and future humanoid robots) in a very natural way using all modalities. The scenario consists of only a limited set of construction elements but offers a rich variety of different tasks. It may serve equally well as the basis for construction experiments in cognitive linguistics (between HCs) and for benchmarking the perceptive, cognitive and manipulative skills of a real-world humanoid robotic system.

Bibliography

Brooks, R. and Stein, L. (1994). Building brains for bodies. *Autonomous Robots*, 1(1):7 – 25.

Grangle, C. and Suppes, P. (1994). *Language and Learning for Robots*. CSLI Publications, Stanford, Ca.

Heydrich, W. and Rieser, H. (1995). Public information and mutual error. Technical report, SFB 360, 95/11, Universität Bielefeld.

Hirzinger, G., Brunner, B., Dietrich, J., and Heindl, J. (1994). Rotex - the first remotely controlled robot in space. In *Proc. IEEE Conference on Robotics and Automation*. IEEE Comp. Soc. Press.

Kawamura, K. and Iskarous, M. (1994). Trends in service robots for the disabled and the elderly. In *Proc. IROS '94 – IEEE/RSJ/GI Int. Conf. on Intell. Robots and Systems*. IEEE Press.

Laengle, T., Lueth, T., Stopp, E., Herzog, G., and Kamstrup, G. (1995). KANTRA - a natural language interface for intelligent robots. Technical report, SFB 314 (VITRA) - Bericht Nr. 114, Universität des Saarlandes.

Meinicke, P. and Zhang, J. (1996). Calibration of a "self-viewing" eye-on-hand configuration. In *Proc. IMACS Multiconf. on Comp. Eng. in Syst. Appl.*, Lille, France.

Moratz, R., Eikmeyer, H., Hildebrandt, B., Knoll, A., Kummert, F., Rickheit, G., and Sagerer, G. (1995). 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.

Restaino, P. and Meinicoff, R. (1985). The listeners: Intelligent machines with voice technology. *Robotics Age*.

Zhang, J., Collani, Y., and Knoll, A. (1997). On-line learning of B-spline fuzzy controller to acquire sensor-based assembly skills. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Albuquerque.