Action-Centered Communication with an Embedded Agent

Jan-Torsten Milde, Kornelia Peters, Simone Strippgen
University of Bielefeld
Department of Linguistics and Literature
P.O. Box 100131, 33501 Bielefeld, Germany
email: {milde, conny, mone}@coli.uni-bielefeld.de

Abstract

In most cases natural language processing is seen as an isolated cognitive capability of a system. Language understanding is often restricted to the mapping of natural language expressions into an internal semantic representation, whereas language production takes an explicit semantic representation as input, from which a natural language utterance is generated. The approach presented in this paper considers the ability to process natural language as a distributed competence of an embedded artificial agent. The agent is able to perceive the environment with its' sensors (vision, tactile, telemetric) and processes natural language directives. In order to describe the ongoing action it is able to produce natural language utterances.

Introduction

In most cases natural language processing is seen as an isolated cognitive capability of a system. Language understanding is often restricted to the mapping of natural language expressions into an internal semantic representation, whereas language production takes an explicit semantic representation as input, from which a natural language utterance is generated. This point of view is inadequate for the use of language in many natural situations of communication.

The approach presented in this paper considers the ability to process natural language as a distributed competence of an embedded artificial agent. The scenario that serves as the system's testbed consists of a simulated assembly robot with an arm-mounted camera. The robot is standing on a work surface, where the assembly parts (coloured wooden bolts, connecting bars and screwing cubes) are scattered. In cooperation with a human interlocutor, the robot can manipulate these construction elements. This agent, named $\mathcal{CoRA}$[2], is able to understand natural language directives as well as to produce natural language utterances concerning the ongoing action (using a subsystem called $\mathcal{RoAD}$[3]).

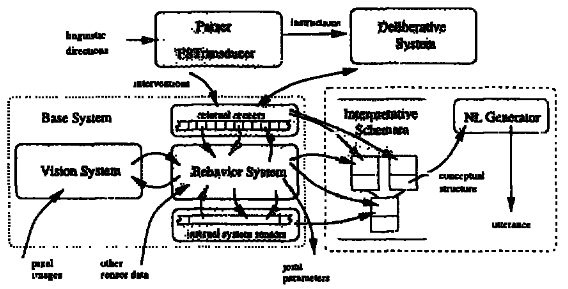


Figure 1: The control architecture of the system. The base system controls the agents' actions, while the deliberative system decomposes action sequences. The FST/parser preprocesses the natural language input. The generating system produces descriptions of ongoing actions.

The architecture

Sensing and acting

The core component of the architecture, which makes the distributed processing of natural language possible, is the so-called base system (see figure 1), which can carry out basic actions autonomously ((Förster, Lobin, & Peters 1995), (Milde 1995) (Milde, Peters, & Strippgen 1997)). The base system consists of a vision system and a behavior-oriented system (s.a. (Brooks 1991)).

In contrast to traditional knowledge-based robot control, control sequences are not based on a detailed world model, but on the perception of the real world. The behavior system is embedded - "situated" - in its environment by means of sensors and actuators, which enable it to detect changes and react to them immediately.

The behavior system contains a hierarchy of behavior modules, each of it specialized for a certain task, which it can schedule and fulfil autonomously. The modularization of the behavior system is motivated by the requirements concerning reactivity and autonomy on the one hand and by the expected user directives on the other hand: All possible linguistic directives must

[2] Communicating Reactive Agent

[3] Robot Action Description

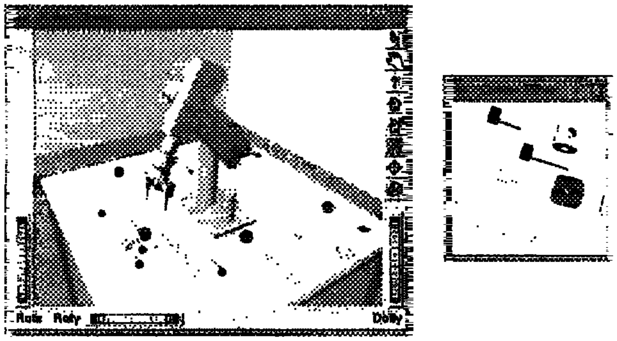


Figure 2: The simulated robot system (6 DOF Puma 260) is positioned on a table. The building parts are scattered on it. Through a hand mounted camera the robot is able to perceive the world. The complete scene as well as the robots' view of the world is displayed to the user.

be depictable to a - as small as possible - set of corresponding behavior modules. The behavior modules are composed of object- or manipulator-oriented behavior routines, which manage the coupling of sensors and actuators. In order to be able to process and carry out action directives on different levels of complexity, the architecture was enlarged by a deliberative system, which models "higher" cognitive competences. This hybrid architecture allows the optimal distribution of the necessary competence and therefore the tasks of the whole system onto both subsystems. The deliberative system is responsible for the sequentialization of complex actions into simple basic actions and schedules the execution of these actions. Suitable feedback from the base system allows the deliberative component to monitor the activity state of the behavior system and feed in the control parameters that are needed for the completion of the next subtask just in time.

Language understanding

The robot can be influenced by typed in action directives. In the hybrid architecture two types of action directives are distinguished: Simple directives - called *interventions* - can manipulate the behavior system directly, complex directives - called *instructions* - influence the deliberative system and only in the second step the behavior system (see also (Goecke & Milde 1998)).

Interventions are fed into the behavior system directly, thus they allow the immediate manipulation of the ongoing behavior ((Peters 1994)). They are processed one by one by a finite state transducer (FST) which recognizes valid input, extracts the relevant information and generates simple attribute-value pairs, which are filled into the internal sensors of the base system. The behavior system is responsible for the situated and time-adequate translation of sensor input into actuator output, treating information from the internal sensors just like any other sensor data. The integration of the different sensor input allows the situated interpretation of directives. As a consequence the processing of elliptical utterances, e.g. situation-dependent object references, which can only be comprehended in the current context of sensing and acting, is made possible.

Head

```
put (AG  a
     OBJ b
     LOC c)
```

Init-State

```
not(pos(b,c))
....
```

.....

Decomposition

```
[GRASP b]
[PUT_DOWN c]
```

Cons-State

```
pos(b,c)
```

Figure 3: The action scheme PUT a b c decomposes the action into a sequence of GRASP b and PUT_DOWN c. Positive feedback of the base system triggers the next action.

Instructions provide resources for planning goals or action sequences and cannot be processed by the base system directly. Therefore instructions are first parsed by a dependency parser, which builds up typed attribute-value pairs (see (Goecke, Peters, & Lobin 1996)). The semantic part of those structures - based on the work by Jackendoff ((Jackendoff 1990)) - is then passed on to the deliberative system, which is responsible for keeping track of long-time goals. The deliberative system uses this semantic part to initialize a corresponding action scheme ((Lobin 1995)). Action schemes contain explicit knowledge about the decomposition of higher level actions into basic actions. An example for an instruction is:

(a) *Put the red cube on the bar.*

Instructions are parsed by a unification-based dependency parser. The semantic representations generated by the parser are passed on to the deliberative component and are then used to choose and initialize so-called action schemes. As a result of the instruction (a) the action scheme PUT is chosen (see fig. 3).

Most important in an action scheme is the decomposition of the complex action into a sequence of basic actions. So the 'put'-action is decomposed into a sequence of a 'grasp'-action and a 'put-down'-action. The accompanying informations (translated into attribute-value pairs) of the first action of this sequence is:

action	:	grasp
det_obj	:	+
obj_colour	:	red
obj_type	:	cube

These values are fed into the internal sensors of the base system (see fig. 4 a). They cause the activation of the behavior module **GRASP** ensuring, together with other sensor data, the situated execution of the grasping of the desired object. After sucsessful completion a positive feedback flows back to the deliberative system and so triggers the next action.

During the execution of a basic action the instructor is allowed to intervene. An example of such an intervention is:

(b) *The other one!*

A correct interpretation of the intervention highly depends on the the current situative context. The intervention is processed by an FST, which generates the following output:

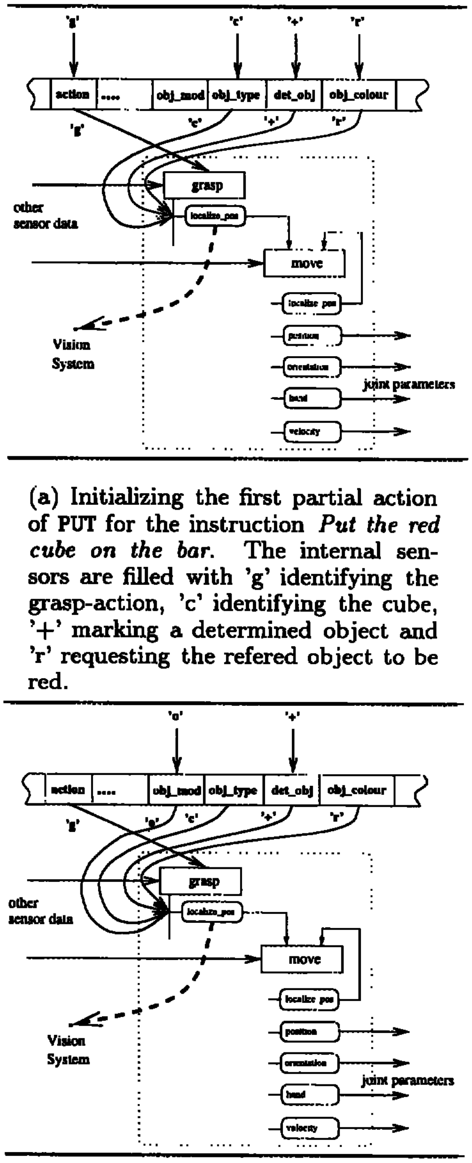
det_obj	:	+
obj_mod	:	other

These values are immediately passed on to the base system, where they are fed into the internal sensors (see fig. 4 b). The missing information about what should be done with this object is extractable from the currently executed action of the robot. When the intervention is uttered the robot is moving towards a red cube in order to grasp it. Therefore the last utterance refers to the 'grasp'-action, which should be continued with the new (*other*) object. In interaction with the vision system, the base system uses the internal sensors' values in conjunction with information on the visually focussed object to identify the object the instructor refered to.

Language generation

Central to the generation process is a data structure called *interpretative scheme* (ISM). Interpretative schemes are a means to encode knowledge about actions and are used to identify action sequences. By this they provide the system with information, which is normally only available to an intelligent external observer of the ongoing scene. In general, concept generation is conceived as a mapping process from subsymbolic, not inherently structured data, to symbolic expressions ("What-to-say"). In *RoAD*, this is accomplished by a hierarchy of *interpretative schemata (ISM)*.

Interpretative schemes consist of three parts: a set of *selection conditions*, a set of *translation rules* and a *conceptual structure* . Basic action concepts are represented on the lowest level of the hierarchy. Higher levels take more complex interpretative schemes, constructed from lower concepts (see fig. 5). Typically, schemata



(a) Initializing the first partial action of **PUT** for the instruction *Put the red cube on the bar.* The internal sensors are filled with 'g' identifying the grasp-action, 'c' identifying the cube, '+' marking a determined object and 'r' requesting the refered object to be red.

(b) Intervening by uttering *The other!* fills the internal sensors for object modification with 'o' and marks the refered object as being determined '+'.

Figure 4: Example of processing a sequence of an instruction (a) and an intervention (b).

positioned lower in this ISM hierarchy represent partial actions of higher ISM. This representation allows the statement of temporal connections as substantiations of actions. In the example (fig.5) the basic ISM GRASP. CARRY, and PUT-DOWN become activated by sensor patterns in the base system. In turn, they serve as selection criteria for PUT. Typically, a sequence of "lower" ISM instantiates a "higher" one. Apart from temporal succession, additional information about objects such as color, size, or position can be propagated to corresponding slots in the conceptual structure.

The sensors *CoRA* currently maintains fall into two classes: *External* and *internal* sensors. The vision system and the contact sensors can be viewed as external sensors in the sense that they require hardware other than mere memory locations. While the contact sensors only have two distinct states (+ or -), the visual sensor can principly handle an infinite number of different objects and object relations. In this case, the processing is computationally much more expensive. In order to maintain real-time behavior of the overall system, visual information is exclusively "offered on demand".

Another source of information for ISM is the activation state of the behavior moduls. A simple example is an activation of the behavior module **grasp**. It represents a selection criterium for the ISM GRASP which in turn contains the conceptual structure **EVENT: grasp, AGENT: i, OBJ:** *obj* etc.

The generation of natural language explanations is a three step process. First the identification of applicable interpretative schemes takes place. Then the translation rules fill the slots of the conceptual structure. This structure will be used by a traditional generator module to generate the natural language output.

The following example shows how an interpretative scheme for "avoid obstacle" is identified and how an appropriate explanation is generated :

(c) *I am avoiding the obstructing object.*

"Avoid obstacle" is recognized, if an intentional, directed movement of the robot is interrupted by the detection of an object, which leads to a movement around the object. The action sequence then continues with the directed movement.

The selection conditions specify a sequence of states which the behavior system has to go through in order to instanciate an interpretative scheme. If this sequence can be identified, the interpretative scheme provides a possible interpretation for the action sequence.

When all conditions are fulfilled the translation rules fill the conceptual structure of the interpretative scheme "avoid obstacle".

The translation rules operate on information of the currently active interpretative scheme and fill the appropriate slots of the conceptual structure. By this, the rules allow the coding of pragmatic knowledge, knowledge about causal relations and about the point of time, when the generation may take place. After filling the conceptual structure the surface generator is able to produce the natural language explanation.

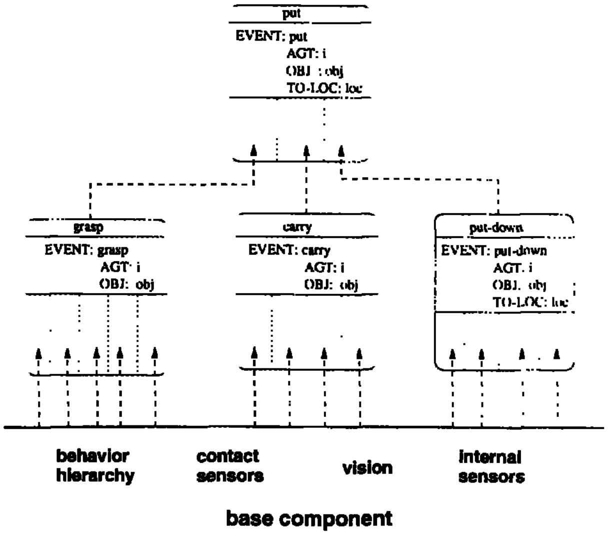


Figure 5: A part of the ISM hierarchy. The basic interpretative schemata grasp, carry and put-down use state and sensor information of the reactive base system. When instantiated information can be transferred to the complex ISM put.

Conclusion

The integration of language, perception and action is the basis for language processing in action-centered communication. In this paper an approach is presented, which considers natural language processing as a distributed competence of an embedded autonomous artificial agent.

The core component of the presented system architecture is a reactive base system, which allows robust interaction with a dynamically changing world. By coupling the base system with the higher deliberative system the robot is able to follow long term goals. Simple directives are processed by the base system directly, while more complex directives are preprocessed by a dependency parser and are then handed over to the deliberative system, which in turn decomposes the complex action into a sequence of basic actions suitable to be executed by the behavior system.

The generation of natural language explanations is based on the status of the behavior system and so uses information about the active behavior module and the current sensor status. This information is integrated by a hierarchy of interpretative schemata, a data structure encoding action knowledge and how this knowledge can be mapped onto conceptual structure. The conceptual structures are the input to the surface generator which produces the natural language utterances of the system.

References

Brooks, R. 1991. Intelligence Without Representation. In *Artificial Intelligence*, volume 47, 139–159.

Förster, S.; Lobin, H.; and Peters, K. 1995. Hybride Architekturen als Grundlage natürlichsprachlicher Steuerung. In Dreschler-Fischer, L., and Pribbenow, S.. eds., *19. Deutsche Jahrestagung für Künstliche Intelligenz, KI-95*. Bielefeld: Gesellschaft für Informatik e.V.

Goecke, K. U., and Milde, J.-T. 1998. Natural language generation in a behavior-oriented robot control architecture. In *submitted to Natural Language Generation, 1998 International Workshop, Canada, Niagara on the lake.*

Goecke, K. U.; Peters, K.; and Lobin, H. 1996. Aufgabenorientierte Verarbeitung von Interventionen und Instruktionen. Report 96/7, Situierte Künstliche Kommunikatoren, SFB 360, Universität Bielefeld.

Jackendoff, R. 1990. *Semantic Structures.* Current studies in linguistics series, 18. Cambridge, MA: MIT Press.

Lobin, H. 1995. *Handlungsanweisungen. Sprachliche Spezifikation teilautonomer Aktivität.* Habilitationsschrift, Universität Bielefeld.

Milde, J.-T.; Peters, K.; and Strippgen, S. 1997. Situated communication with robots. In *Proceedings of the first international Workshop on Human-Computer Conversation, Bellagio, Italy.*

Milde, J.-T. 1995. A hybrid control architecture for a simulated robot manipulator. In *Proceedings of the 13th IASTED International Conference on applied informatics.*

Peters, K. 1994. Natürlichsprachliche Steuerung eines behaviorbasierten Roboters. Report 94/8, Situierte Künstliche Kommunikatoren, SFB 360, Universität Bielefeld.