Proceedings of the 2000 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

Human-Robot Interaction through Spoken Language Dialogue

L. Seabra Lopes and **A. Teixeira**

Departamento de Electrónica e Telecomunicações
Universidade de Aveiro / P-3810 Aveiro - Portugal

Abstract

The development of robots able to accept, via a friendly interface, instructions in terms of the concepts familiar to the human user remains a challenge. It is argued that designing and building such intelligent robots can be seen as the problem of integrating four main dimensions: human-robot communication, sensory motor skills and perception, decision-making capabilities and learning. Although these dimensions have been thoroughly studied in the past, their integration has seldom been attempted in a systematic way. It is further argued that, for the common user, the only sufficiently practical interface is spoken language. The "body and soul" of Carl, a robot currently under construction in our lab, are presented. The spoken language interface is given particular attention.

1. Introduction

The development of robots that don't have to be programmed in the classical way and, instead, can accept instructions at the level of concepts of the human user will be a major breakthrough. If a flexible manufacturing system is supposed to produce a variety of products and in small quantities, then industrial robots will tend to play the role of craftsmen. Both service robots and flexible industrial robots will need to use sensors extensively in order to develop a high level understanding of their tasks.

Robot decision-making at the task level is, therefore, a central problem in the development of the next generation of robots [10,23,24,26]. As the modularity and reconfigurability of the hardware are enhanced, the number of action alternatives at the task-level increases significantly, making autonomous decision-making even more necessary.

The development of task-level robot systems has long been a goal of robotics research. It is of crucial importance if robots are to become consumer products. The use of the expression *task-level* is due to Lozano-Perez *et al.* [17]. The idea, that was already present in automatic robot programming languages, such as AUTOPASS and LAMA, developed in the 1970's, has been taken up in recent years by other researchers [10,23].

The authors of the present paper are currently involved in CARL, a project aimed at contributing to the development of task-level robot systems.

This paper focuses on the human-robot interface. The main claim is that the only acceptable user interface for a task-level robot is a spoken language interface. As will be explained, no other form of interface is flexible enough to solve the symbol grounding problem in a way that makes the robot system practically useful.

The paper is organized as follows. Sections 2 and 3 present the motivations for the CARL project, its reference architecture and some design principles we have adopted. Sections 4 and 5 present conceptual work as well as some development on the spoken language interface. Section 6 describes current experimental work. Finally, conclusions are presented.

2. The CARL Project

The viewpoint in early artificial intelligence research was to evaluate an agent's intelligence by comparing it's thinking to human-level thinking. The development of human-level intelligence [28] is probably a too ambitious goal for the current state of art. We believe that it is more reasonable to develop useful robotic systems with hardware and intelligence tailored for specific applications. This will provide experience on how to integrate different technologies and execution capabilities and, eventually, will enable us to scale up to more general robot architectures.

Currently, the major effort involved in developing useful intelligent robots is, we believe, in the integration of different capabilities.

The authors are currently involved in a project titled "*Communication, Action, Reasoning and Learning in robotics*" (CARL). The activities started in July 1999.

CARL is based on the hypothesis that a combination of reactivity with reasoning is more likely to produce useful results in a relatively near future than the purely reactive or behavior-based approaches. This is especially true for robots that are expected to perform complex tasks requiring decision-making.

The integration of reactivity with reasoning has proved to be difficult to achieve. Traditional architectures have focused on traditional problems like reasoning, representation, and NLP and alternative architectures have focused on problems such as real-time perception and motor control. There have been few, if any, satisfying attempts to integrate the two. The position (and driving hope) of the CARL project is that most of the encountered difficulties are the result of not addressing properly the learning and, especially, the interface issues.

In the traditional approach to building intelligent systems, the human devises a formal language and uses it to specify the needed representations of the world. As the

0-7803-6348-5/00/$10.00 ©2000 IEEE.

application becomes more and more complex, the programmer’s task becomes overwhelmingly difficult. Automatic programming languages, embedding various planning capabilities, have been developed in order to simplify the programming problem. Programming by human demonstration [13,20] and learning techniques [20] have been used for the same purpose. None of these approaches solved the problem. Robot programming is the bottleneck where robot development gets stuck. However, this only hides the more fundamental symbol grounding problem [11,18].

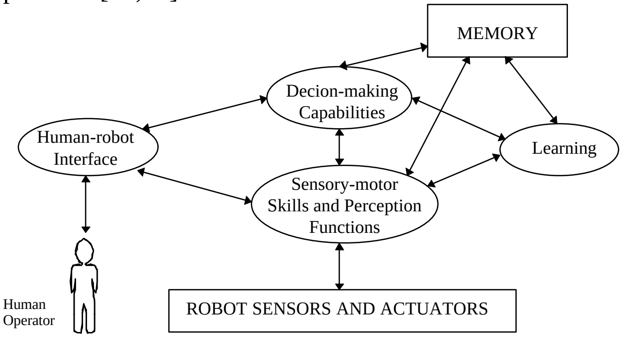
MEMORY
Decion-making Capabilities
Human-robot Interface
Learning
Sensory-motor Skills and Perception Functions
Human Operator
ROBOT SENSORS AND ACTUATORS

Fig. 1 - Reference architecture for the CARL project

Sometimes it is argued that symbol grounding should be a bottom-up process. However, like humans, machines will benefit from using both grounding directions, i.e. symbols externally communicated will be grounded in a top-down fashion while other symbols will be discovered in a bottom-up fashion. Symbol grounding is intimately connected to learning: supervised learning is the most promising approach for top-down grounding while clustering is appropriate for bottom-up grounding.

To correctly address symbol grounding in the context of task execution, the first thing to notice is that most symbols are inherent to the tasks. In that case, the human user, who defines the tasks, will be a primary source of information for the symbol grounding process. The human will be simultaneously the user and the teacher. The communication interface between human and robot is, therefore, of primary importance.

If we are developing intelligent robots with significant decision making capabilities, the use of spoken natural language seems unavoidable. For sure, this is a comfortable interface for humans. But, it is unavoidable because no other alternative is practical enough. The common (naïve) user does not want to learn a formal programming syntax. To our knowledge, the only project that has been consistently working in this direction is the JIJO-2 project [2,9].

Teaching a task, for instance, should be an interactive process, in which the human explains one or two steps of the task, the robot tries them out and then the human explains a little more, and so on [12]. Natural language seems also to be the best way for easily guiding the robot in recovering from failures. In general, natural language seems to be the only practical way of presenting new symbols (representing physical objects, failure states, operations, or whatever) to the robot. Grounding these symbols depends essentially on the robot sensors and learning abilities.

Learning, defined as the process by which a system improves its performance in certain tasks based on experience, is one of the fundamental abilities that a system must possess in order to be considered intelligent [19]. The major contribution of learning is to extract new knowledge from representative cases or episodes when such knowledge cannot practically be designed and programmed from scratch.

Until now, research in applying learning in robotics and automation has focused on learning control functions, adaptation and dealing with uncertainty. Behavior-based robotics has investigated the integration of learning mechanisms in robot architectures for navigation and environment exploration [1]. Learning at the higher levels of the robot architecture, namely in planning and failure recovery, is also necessary [24].

The goal of CARL is, therefore, to study the integration of, not only reasoning and reactivity, but also learning and human-robot interaction (fig. 1). In this paper, the interface dimension of the problem is given special attention.

3. The Carl robot: "Body and Soul"

Carl is the name of the robot of the CARL project. It is currently under construction.

3.1. Mobile platform

Carl is based on a Pioneer 2-DX indoor platform from ActivMedia Robotics, with two drive wheels plus the caster. It includes wheel encoders, front and rear bumpers rings, front and rear sonar rings and audio I/O card. The platform configuration that was purchased also includes a micro-controller based on the Siemens C166 processor and an on-board computer based on a Pentium 266 MHz with PC104+ bus, 64 Mb of memory and a 3.2 Gb hard drive. The operating system is Linux.

For the speech interface, we are using an LVA-7280 digital microphone array from Labtec. We are currently installing a compass (Vector 2XG from Precision Navigation Inc.) and a PTZ104 PAL Custom Vision System.

With this platform, we hope to be able develop a completely autonomous robot capable, not only of wandering around, but also of taking decisions, executing tasks and learning.

3.2. "Innate" capabilites

Our idea for Carl is to integrate, in the construction phase, a variety of processing and inference capabilities. In contrast, the initial body of knowledge will be minimal. After this phase is concluded (after the robot is born!), a life-long learning process can start. Carl learns new skills, explores its environment, builds a map of it, all this with frequent guidance from the human teacher.

Some of the "innate" capabilities / knowledge, that will be integrated in Carl during the construction phase are:

- Wandering around in the environment while avoiding obstacles; this is the only "innate" behavior.
- Natural language processing (see section 4.2), supported by a fairly comprehensive vocabulary of English words; the meanings of most words are initially unknown to Carl.
- Basic speech processing (see section 4.3).
- A small dictionary of words and their meanings for identifying the robot's sensors and basic movements; these are the initially ground symbols over which Carl will incrementally build his knowledge.
- Ontologies for organizing and composing behaviors, map regions, dialogues, task plans, episodic memory, etc.
- Knowledge of basic mathematical functions, that the teacher can use for teaching new concepts or behaviors.
- Logical deduction (in a Prolog framework)
- Capabilities for task planning and execution monitoring.
- Capabilities for learning numerical functions.
- Capabilities for learning symbolic classification knowledge.
- Capabilities for explanation-based learning and case-based reasoning.

Part of these capabilities can be implemented by adapting prototypes previously developed by the research team (Seabra Lopes, 1997 and 1999ab). In section 6, an example of a teaching dialogue will be presented.

4. The Spoken Language Interface

An important component of the Carl robot, that deserves special attention in this paper, is the spoken language interface (fig. 2). This section discusses requirements, hardware to integrate in Carl and software tools.

4.1 General

The interface has two main components: (1) Natural Language Processing, and (2) Speech Processing. The first, handles language understanding with lexical analysis, grammar rules for sentence parsing. Based in task knowledge and human inputs, it also generates the information to convey to the user. The second component, provides recognition results (in the form of sentences or word lattices) to the first module; and transmits messages using speech synthesis to the human user.

The process of message generation by the robot can be conceptually split in three phases [16]: 1. A Conceptualizer generates preverbal messages consisting of conceptual information whose expression is the mean for realizing the robot's "intentions"; 2. A Formulator uses a grammar (and possibly other information) to generate a concrete message; and 3. An Articulator transforms the message into an acoustic wave.

In Carl's architecture, the Conceptualizer will be integrated in the Dialogue Manager, the Formulator will be part of the Natural Language Processing module, and the Articulator will be part of the Speech Processing module.

As the human speaks, the Speech Recognition module extracts a natural language (NL) sentence and sends it to the Natural Language Understanding (NLU) module for processing. In turn, NLU extracts from the NL sentence a formal message in the HRCL language (see section 5). This formal message is then processed by the Dialogue Manager, that eventually provides a response. The response, produced by the Conceptualizer, is also a message in the HRCL language.

A mixed initiative dialogue has several advantages compared to a master-slave dialogue. On one hand, the dialogue becomes more natural. Furthermore, the robot may need to ask questions [2].

An important issue is keeping track of dialogue context. For that purpose, a *hierarchy of dialogue contexts* will be used. A set of performatives in the HRCL language are related to this (see section 5). The "task hierarchies" of [8] are similar to our hierarchies of dialogues.

A dialog with Carl can be about the execution of a task in his environment. A sub-dialogue of that dialogue can be about a particular action that the robot does not know how to perform. A sub-sub-dialogue can be about some particular feature of the environment to which Carl should pay attention while performing the action.

4.2. Natural Language (NL) Processing

The CPK NLP suite (Brondsted, 1999ab) is being used for NL Understanding. The C API provides mechanisms for loading external grammar files, activating and deactivating subgrammars and for performing parsing.

In order to understand under-specified sentences, the system needs to keep track of salient information. Examples of salient information are people, objects and events being talked about. A mechanism based in the "centering" theory of linguistics has been applied in the JIJO-2 project [9].

There are several interesting aspects to discuss concerning NL Generation. The Formulator module receives as input an HRCL message from Carl's Dialogue Manager, produces a semantic frame and, finally, a NL sentence.

Processes available for the Formulator range in sophistication from inflexible canned methods to maximally flexible feature combination methods [7]. Canned systems simply print a string of words without any change. Template systems are the next level of sophistication, followed by phrase based systems employing what can be seen as generalized templates. Feature based systems are the limit point of sophistication. We consider the template approach, mostly used for single-sentence generation, appropriate for our

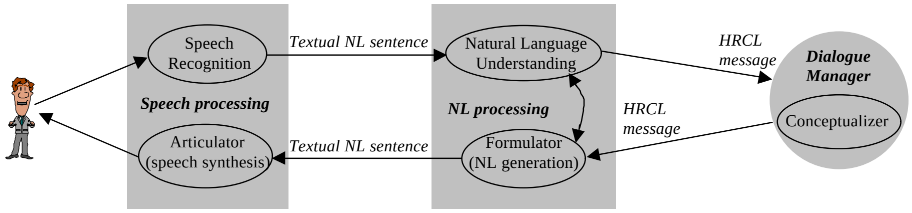

Fig. 2 - Spoken language interface

application, since linguistics grammars provide well-specified collections of phrase structure rules.

To improve the communication, messages should vary incorporating randomness in the choice of each message component. This can be done using a random message generator guided by a grammar as used in the CPK NLP suite SGEN programs [3].

An audio interface requires the short term memory of the listener. The capacity of memorizing, however, is rather limited, therefore messages can't convey too much information at a time. Unlike graphical interfaces, a speech-only interface is not persistent. Functionality of the application is hidden, and the boundaries of what can and cannot be done are invisible [29]. Techniques such as: incremental and expanded prompts; tapering, shortening the interactions as user gains experience; and hints, can be used to make the interaction more natural. For doing this, the Formulator needs to keep track of past messages used in the current conversation.

Other question that can be addressed in the Formulator is to give implicit feedback to the user about speech recognition results. Because we have far from perfect recognition of speech (see section 6), it is in some occasions useful to transmit to the user what the system recognized. For example, the robot is standing near the stairs and the user tells him to turn back, but due to misrecognition, the system understands the command to go forward. But then, using knowledge and sensor information, the robot "decides" for the need of explicit confirmation asking the user to confirm the order. This is like a sub-conscious reaction to a danger situation. If the robot doesn't include in the confirmation message what was the perceived order, the user as no way of knowing what is being confirmed.

This mechanism can use the confidence measure of the recognition process. In a dialogue system, ARISE [27], if the confidence is high, implicit confirmation is applied; else explicit confirmation is used.

4.3. Speech Processing

With respect to the Speech Recognition module, the main requirement is that the whole process looks natural. This implies handling continuous speech, being speaker-independent without need for training, handling natural speech phenomena such as hesitations, running in near real-time, and even allowing for interruptions when the robot is speaking [5]. The dialogue manager can help speech recognition by sending context information to guide the use of dynamic grammars.

A flexible system allows users to speak the same commands in many different ways. But, the more flexibility an application provides for user input, the more likely errors are to occur. A compromise is needed.

In conversations, timing is critical. People give meaning to pauses. For example, users may reply to a prompt and then not hear an immediate response, leading them to repeat the response.

Within CARL, the Entropic grapHvite [22] is being used for implementing the recognition. The CPK NLP suite [3,4] can also be used in developing finite state grammar networks used by grapHvite.

If noise conditions prove to be very adverse for the grapHvite recognizer (that is trained in clean speech), development of a new recognizer can be done using HTK [30] and easily used in the system. Preliminary studies show degradation when there is a mismatch in training and test speech conditions.

The Articulator module will be responsible for speech synthesis. Besides doing the actual text-to-speech (TTS) work, this module will be responsible for conveying paralinguistic information, such as emotion [21]. It is our belief that even rudimentary approaches will be appreciated by end users.

Extra information, provided by the NL generation module, regarding prosody and emotion to be conveyed by speech, can reduce the complexity of the needed text analysis tasks.

Synthetic speech needs to be intelligible and as natural as possible; manipulation of speed, volume and pitch should be possible; provide good text normalization; a good API should be available; support for Linux and possibly Windows operating systems. We are currently using IBM ViaVoice Outloud, not only because it matches the requirements, but also because it is a free open source system with which we have some previous experience.

5. Human-Robot Communication Language

As it was pointed out, spoken natural language dialogue is the only practical way a non-expert user has for specifying and teaching a task to a robot.

To implement this type of communication, the robot will have to be able to generate as well as to interpret spoken natural language sentences.

Table I - HRCL Performatives (S = sender; R = receiver; C = content)

Performative	Meaning
ask(S,R,C)	S wants R to provide one instantiation of sentence C
ask_if(S,R,C)	S wants to know if R thinks sentence C is true
tell(S,R,C)	S thinks sentence C is true and tells that to R
deny(S,R,C)	S does not know if sentence C is true and tells that to R
insert(S,R,C)	S asks R to consider sentence C true
delete(S,R,C)	S asks R to no longer consider sentence C true
achieve(S,R,C)	S asks R to perform action C in its physical environment
error(S,R)	S informs R that S cannot not understand R's previous message
sorry(S,R)	S informs R that S understands R's previous message but cannot provide a response
standby(S,R,C)	S wants R to announce its readiness to provide response to message C and standby
ready(S,R)	S is ready to respond to a message previously sent by R
next(S,R)	S wants R's next response to a message previously sent by S
rest(S,R)	S wants R's remaining responses to a message previously sent by S
discard(S,R)	S does not want R's remaining responses to a message previously sent by S
register(S,R)	S announces its presence to R
dialogue(S,R,C)	S proposes to R a (sub-)dialogue about subject C
dialogue_accept(S,R)	S accepts to participate in a (sub-)dialogue previously proposed by R
dialogue_reject(S,R)	S rejects to participate in a (sub-)dialogue previously proposed by R
dialogue_end(S,R,C)	S proposes to R to end a (sub-)dialogue about subject C
dialogue_end_accept(S,R)	S accepts to end a (sub-)dialogue as proposed by R
dialogue_end_reject(S,R)	S rejects to end a (sub-)dialogue as proposed by R

Internally, of course, the exchanged messages are represented more formally. The multi-agent systems community has been developing languages for communication between agents. Probably the best known is ACL, acronym of *Agent Communication Language* [15]. ACL is composed of three parts: an ontology for a given domain; an inner language, for knowledge representation; and an outer language and protocol for information and knowledge exchange. The outer language is KQML, or *Knowledge Query and Manipulation Language*. It offers a variety of message types, called performatives. The inner language is KIF, or *Knowledge Interchange Format*, a generic knowledge representation language. An ACL message is therefore a KQML performative in which the arguments are KIF sentences formed from words in the ACL vocabulary.

For representing the messages exchanged between the robot and its human user/teacher we took some inspiration in ACL. In development is HRCL, our *Human-Robot Communication Language*. For knowledge representation within the interchanged messages, we use plain Prolog, since this language is also being used to implement the decision-making modules as well as for knowledge representation. The domain ontology is being defined taking into account the kind of world that we anticipate the robot will be able to “see” (given the sensors that we have installed) and the kind of capabilities we want do develop. The message types are mostly inherited from KQML.

In order to constrain the problem, a system with only one learning robot and only one human teacher will be considered. This means that KQML performatives related to networking are not necessary and therefore won't be included in HRCL (at least not in the first phase of the project). Some performatives for start and termination of dialogues and sub-dialogues were considered. The idea is that the robot will keep track of context with help of a hierarchy of dialogues and sub-dialogues. Table I lists the most important HRCL performatives. There are also performatives for canceling, undoing or reversing the effects of previous performatives (for clarity of presentation, these are not listed in the table).

6. Current Work

In this section, we describe experiments with the spoken language interface. We intend to use this type of interface both for learning behaviors and task-level knowledge.

The interaction of the robot with the environment and, in particular, the execution of tasks is supported by a set of sensory-motor skills, each of them involving a tight connection of perception to action [20]. A classical example, from the robotized assembly domain, is the peg-in-hole insertion skill. In a mobile robot, navigation with obstacle avoidance is also a basic skill.

Learning a behavior or basic skill is, in most cases, a problem of learning a numerical function [20]. The application of various types of neural networks, due to their ability to handle non-linearity and uncertainty, has already delivered interesting results. Another alternative is regression trees. The use of reinforcement learning in on-line skill learning or refinement is also being investigated.

There is, however, a major problem that has not been solved: training data collection and pre-processing. The usual approach is to carry out intensive and highly tiring training sessions during which the training data is collected in a more or less manual way. For service robots, collecting training data must be an incremental process that seems natural to the user. Another phase is data pre-processing [25]. Here, the space of input variables (features) is transformed in order to produce other features that are more informative and therefore enable more efficient learning. Sometimes, without this transformation, learning is completely impossible.

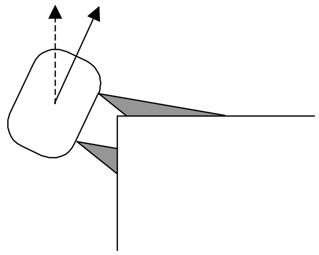

Fig. 3 - Robot going around a corner

Suppose we want to teach a robot how to go around a corner (Fig. 3). This robot has a compass and also two sonar sensors on his right-hand side. The orientation provided by the compass is not really relevant for learning the behavior. What is important is the orientation of the robot relative to the corner. Of course, if many examples are provided, the learning algorithm will eventually be able to generate the control function. But if the right feature (for instance the orientation relative to the initial wall) is provided, much less examples will be required. The human teacher should provide this kind of information to the robot.

Table II shows a possible dialogue during which the robot learns how to go around a corner. In the end, the meaning of the symbol "Going around a corner" is defined.

Table II - An example of a teaching dialogue

T: *I'll teach you how to go around a corner*.
R: *Hum, going around a corner …*
T: *The corner is on your right hand side*.
R: *Ok, corner on the right*.
T: *Don't forget your present heading. Call it initial heading*.
R: *Ok*.
T: *And, pay attention to the difference between the heading and the initial heading*.
T: *Now, start moving forward*.
R: *Ok, moving forward*.
T: *Turn a little to your right.*
R:
T: *Reduce rotation.*
R:
T: *No rotation.*
R: *Moving forward only.*
T: Stop!
R: It's done?
T: Yes!

A simple mobile robot can have behaviors like following walls, moving to walls, going around corners, etc. A first version of the instruction-based behavior learning mechanism is currently operational in Carl. After the training, behavior synthesis is done by a standard back-propagation neural network.

This style of interaction is equally useful for teaching task-level knowledge. To start with, as already demonstrated by the JIJO-2 project [2,9], the human can help the robot to build a map of its environment. Furthermore, the human can give explanations, outline plans, etc., that later the robot can adapt to new situations [23,24]. For this, explanation-based learning and case-based reasoning are basic capabilities that must be included.

One application we are currently addressing is the robotic "tour guide". In this case the robot, ideally, enters in dialogue with the tourist or visitor, answering the questions that the tourist asks, and skipping the aspects in which the tourist does not seem to be interested. That means that the robot must really maintain a knowledge representation about what there is to see, adapt to the user and, at the same time, do all other things that robots are expected to do in the physical world, particularly navigation and localization. A robotic tour guide for a museum in Bonn has been built at Carnegie-Mellon University [6], but this robot only has pre-recorded messages and does not enter any kind of dialogue.

We therefore want to see how far dialogue can go in such application with available speech technology. Preliminary tests were performed using the already described setup (LVA-7280 microphone, grapHvite and the CPK NLP suite). A NL grammar of 30 rules and a vocabulary of 66 words was used to represent sentences from the following HRCL performatives: achieve, tell, ask, ask_if, register and sorry (see Table I).

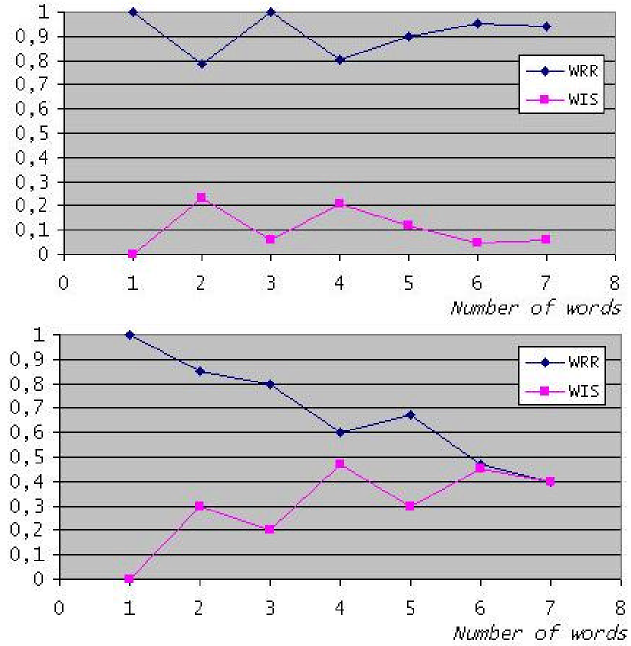

Fig. 4 - Speech recognition results: near the microphone (top) and at 2 m from the microphone (bottom).

Many sentences (nearly 100) were spoken near the microphone and the recognition results were analyzed. Fig. 4a plots the Word Recognition Rate (WRR) and the Word Insertion Rate (WIR) versus the number of words in the original sentence. These results are acceptable.

The experiment was then repeated for a distance of 2m from the microphone (fig. 4b). For now, these results seem to indicate that digital microphone array technology is still not able to support natural language dialogues with robots.

7. Conclusion

The CARL project aims to contribute to the development of task-level robot systems by studying the interrelations and integration of what were identified as the four major dimensions of the problem: *human-robot interaction*, *sensory-motor skills and perception*, *decision-making* and *learning*. The lack of integration efforts in robotics, observed in the past, is explained by various technological limitations. However, thanks to the increasing availability of compact hardware at reasonable cost and to the dramatic increase in computational power that was observed in recent years, the proposed integration work is now becoming feasible. The hardware and the intended capabilities of the Carl robot were described. The idea is that the robot is "born" with minimal knowledge but plenty of capabilities for learning.

This paper focused on the human-robot interface. This is of top importance, not only for task specification but also for teaching new skills and new task-level knowledge. Ultimately, solving the symbol grounding problem depends on combining learning and human-robot interaction in an appropriate way. It was argued that, for the common user of the future intelligent service robots, there is no other alternative than basing the interface on spoken language.

At the current state of affairs, it is already possible to teach the robot simple basic behaviors through natural language. For representing formally the spoken messages exchanged between human and robot, we propose the HRCL language, that is inspired in ACL. Experimental results seem to indicate that the digital microphone array technology is still not able to support spoken language dialogues with robots. We are, currently, looking for alternatives for the speech interface.

Acknowledgements

CARL is funded by the Portuguese research foundation (FCT), reference PRAXIS/P/EEI/ 12121/1998. The authors thank the contributions of P. Fonseca, K.L. Doty, F.Vaz, N.M Ferreira and N.F. Ferreira.

References

[1] Arkin, R.C. (1998) *Behavior-Based Robotics*, MIT Press.

[2] Asoh, H., S. Hayamizu, I. Hara, Y. Motomura, S. Akaho and T. Matsui (1997) «Socially Embedded Learning of the Office-Conversant Mobile Robot Jijo-2», *Proc. 15th Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, Nagoya, p. 880-885.

[3] Brondsted, T. (1999a) «The Natural Language Processing Modules in REWARD and IntelliMedia 2000+», *LAMBDA 25*, Copenhagen 1999, pp. 91-108.

[4] Brondsted, T. (1999b) «The CPK NLP Suite for Spoken Language Understanding», *Proceedings of Eurospeech'99*, Budapest, Hungary.

[5] Bronsdsted, T., et al (1998) *A Platform for developing Intelligent Multimedia Applications*, Technical Report R-98-1004, Center for PersonKommunikation (CPK), Aaalborg University.

[6] Burgard, W., et al. (1999) Experiences with an Interactive Museum Tour-Guide Robot, *Artificial Intelligence*, 114, 3-55.

[7] Cole, R.A., et al [edrs.] (1997) *Survey of the State of the Art in Human Language Technology*, Cambridge University Press.

[8] Ehrlich, U. (1999) «Task Hierarchies Representing Sub-Dialogs in Speech Dialog Systems», *Proceedings of Eurospeech'99*, Budapest.

[9] Fry, J., H. Asoh and T. Matsui (1998) «Natural dialogue with the JIJO-2» ofiice robot, *Proceedings of the IROS'98*.

[10] Haigh, K.Z. and M. Veloso (1996) «Interleaving Planning and Robot Execution for Asynchronous User Requests», *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS'96), Osaka, Japan, pp. 148-155.

[11] Harnad, S. (1990) «The Symbol Grounding Problem», *Physica D*, vol. 42, pp. 335-346.

[12] Huffman, S.B. and J.E. Laird (1993) «Learning Procedures from Interactive Natural Language Instructions», *Proc. 10th International Conference on Machine Learning*, Amherst, MA, pp. 143-150.

[13] Kang, S.B. and K. Ikeuchi (1995) «Toward Automatic Instruction from Perception: Recognizing a Grasp from Observation», *IEEE Transactions on Robotics and Automation*, vol. 11, p. 670-681.

[14] Khatib, O. (1999) "Mobile Manipulation: the Robotic Assistant", *Robotics and Autonomous Systems*, 26 (2-3), 175-183.

[15] Labrou, Y., T. Finn and Y. Peng (1999) "Agent Communication Languages: the Current Landscape", *IEEE Intelligent Systems*, 14 (2), 45-52.

[16] Levelt, W.J.M. (1989) *Speaking - From Intention to Articulation*, ACL-MIT Press Series in Natural-Language Processing, The MIT Press.

[17] Lozano-Pérez, T., J.L. Jones, E. Mazer and P.A. O'Donnell (1989) «Task-level Planning of Pick and Place Robot Motions», *Computer*, vol. 22, n.3 (March), pp. 21-29.

[18] MacDorman, K.F. (1999) "Grounding Symbols through Sensorymotor Integration", Journal of the Robotics Soc. of Japan, 17 (1), 20-24.

[19] Mitchell, T.M. (1997) *Machine Learning*, WBC/McGraw-Hill.

[20] Morik, K., M. Kaiser and V. Klingspor [edrs.] (1999) *Making Robots Smart. Behavioral Learning Combines Sensing and Action*, Kluwer Acedemic Publishers.

[21] Olive, J.P. (1997) «"The Talking Computer": Text to Speech Synthesis», *HAL's Legacy: 2001's Computer as Dream and Reality*, David Stork (ed.), The MIT Press.

[22] Power, K., R. Morton, C. Matheson and D. Ollason (1997) *The grapHvite Book For grapHvite v1.1 Entropic*, Cambridge Research Laboratory.

[23] Seabra Lopes, L. (1997) *Robot Learning at the Task Level: A Study in the Assembly Domain*, Universidade Nova de Lisboa, Ph.D. Thesis.

[24] Seabra Lopes, L. (1999b) «Failure Recovery Planning in Assembly Based on Acquired Experience: Learning by Analogy», *Proc. IEEE. Int. Symp. on Assembly and Task Planning*, Porto, Portugal.

[25] Seabra Lopes, L. and L.M. Camarinha-Matos (1998) «Feature Transformation Strategies for a Robot Learning Problem», *Feature Extraction, Construction and Selection. A Data Mining Perspective*, H. Liu and H. Motoda (edrs.), Kluwer Academic Publishers.

[26] Seabra Lopes, L. And A. Teixeira (2000) Teaching Behavioral and Task Knowledge to Robots through Spoken Dialogues, *My Dinner with R2D2: Working Notes of the AAAI Spring Symposium on Natural Dialogues with Practical Robotics Devices*, Stanford, CA.

[27] Sturm, J., E. den Os and L. Boves (1999) «Issues in Spoken Dialogue Systems: Experiences with the Dutch ARISE System», *ESCA Tutorial and Research Workshop (ETRW) Interactive Dialogue in Multi-modal System (IDS)*.

[28] Turing, A.M. (1950) «Computing Machinery and Intelligence», *Mind*, 59, p. 433-460.

[29] Yankelovich, N. (1996) «How do users know what to say?», *Interaction*, vol 3, p. 32-43.

[30] Young, S., J. Odell, D. Ollason, and P. Woodland (1999) *The hTk Book Version 2.2*, Entropic Ltd.