2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Progress in Programming the HRP-2 Humanoid Using Spoken Language

Peter Ford Dominey, Anthony Mallet, Eiichi Yoshida

Abstract—**The current research analyses and demonstrates how spoken language can be used by human users to communicate with the HRP-2 humanoid to program the robot's behavior in a cooperative task. The task involves the humans and the HRP-2 working together to assemble a piece of furniture. The objectives of the system are to 1. Allow the human to impart knowledge of how to accomplish a cooperative task to the robot, i.e. to program the robot, in the form of a sensory-motor action plan. 2. To do this in a semi-natural and real-time manner using spoken language. In this framework, a system for Spoken Language Programming (SLP) is presented, and experimental results are presented from this prototype system. In Experiment 1, the human programs the robot to assist in assembling a small table. In Experiment 2, the generalization of the system is demonstrated as the user programs the robot to assist in taking the table apart. The SLP is evaluated in terms of the changes in efficiency as revealed by task completion time and number of command operations required to accomplish the tasks with and without SLP. Lessons learned are discussed, along with plans for improving the system, including developing a richer base of robot action and perception predicates that will allow the use of richer language. We thus demonstrate - for the first time - the capability for a human user to tell a humanoid what to do in a cooperative task so that in real time, the robot performs the task, and acquires new skills that significantly facilitate the cooperative human-robot interaction.**

I. Introduction

Humanoid robots are now physically capable of locomotion, object manipulation, and an essentially unlimited set of sensory motor behaviors. This sets the scene for the corresponding technical challenge: How can non-specialist human users interact with these robots for human robot cooperation? Crangle and Suppes [1] stated: "(1) the user should not have to become a programmer, or rely on a programmer, to alter the robot's behavior, and (2) the user should not have to learn specialized technical vocabularies to request action from a robot." In this context spoken language provides a very rich vector for communication. As the meaning to be communicated becomes more complex, so do the grammatical constructions used. Construction grammar (CxG) provides a linguistic formalism for achieving the required link from language to meaning [2]. Meaning is represented in a predicate-argument (PA) structure as in (2), based on generalized abstract structures as in (3). The power of these constructions is that they are based on abstract "variables" that can take an open set of arguments .

(1) John put the ball on the table.
(2) Transport(John, Ball, Table)
(3) Event(Agent, Object, Recipient

We can thus use the PA structures to extract robot commands from natural language, and to generate natural language descriptions of physical events extracted from video scenes [3-8]. The objective of the current research is to begin to use natural language in order to allow human users to "tell the robot what to do" and "teach it" or "program it" with spoken language. In a related context, Nicolescu and Mataric [9] employed spoken language to allow the user to clarify what the robot learned by demonstration. In order to explore how language can be used more directly, Lauria et al. [10] asked naïve subjects to provide verbal instructions to a robot in a visual navigation task. Their analysis of the resulting speech corpora, yielded a set of verbal action chunks that could map onto robot control primitives. They demonstrated the effectiveness of such instructions translated into these primitive procedures for actual robot navigation [11]. This indicates the importance of implementing the mapping between language and behavioural primitives for natural language instruction or programming [see 12]. Learning by imitation and/or demonstration likewise provide methods for humans to transmit desired behaviour to robots [13-14]. The current study extends such methods in a complimentary way.

For the first time, spoken language is used in real-time to allow the user to command a bi-manual humanoid, and to create new behavioural patterns that can be immediately re-used based on ongoing task requirements. To do this we must first determine the set of action/command primitives that satisfy two requirements: 1. They should allow a

Manuscript received January 30, 2007. Supported in part by French Minister of Research under grant ACI-TTT, the LAFMI, and the CNRS Joint Robotics Laboratory, LAAS Toulouse. P. F. Dominey is with the CNRS, 67 Bd Pinel 69675 Bron Cedex, France (phone: 33-437-911266; fax: 33-437-9112110; e-mail: dominey@ isc.cnrs.fr). A. Mallet is with the LAAS, CNRS, Toulouse France (e-mail: anthony.mallet@ laas.fr). E. Yoshida, is with AIST-CNRS Joint French-Japanese Robotics Lab (JRL) National Institute of Advanced Industrial Science and Technology (AIST) 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 Japan, and is Co-Director of the CNRS/AIST French-Japanese Joint Robotics Laboratory, with the LAAS, CNRS, Toulouse France (e-mail: eiichi.yoshida@ laas.fr).

logical decomposition of the task into units that are neither too small (i.e. move a single joint) nor too large (perform the whole task). 2. They should be of general utility so that other tasks can be performed with the same set of primitives.

II. A SCENARIO FOR HUMAN-ROBOT COOPERATION

A. The Scenario

Figure 1 illustrates the HRI scenario that we analyze in this research which involves two humans and the HRP-2 cooperating in the construction of a small table. The construction task will involve attaching the legs to the surface of the table with wood screws. User1 on the left interacts with the robot and with User2 on the right via spoken language.

User1 will command the robot to prepare to receive one of the table legs that will be passed to it by User2. The robot waits until it receives a "continue" signal from User1, and will then pass the leg to User1 who will take the leg, and then ask the robot to hold the table top steady, allowing User1 to attach the leg to the table. User1 then tells the robot to release the table. At this point, the first leg has been attached to the table, and the "get the leg and attach it" sequence can be repeated.

B. On-line commanding with repetitive a subsequence

On-line commanding allows the user to be responsive to new situations, and to learn him/herself by taking the robot through a given task or tasks. On the other hand, for tasks that are well defined, the user should be able to program the robot by saying the sequence of commands and storing it before the actual execution. In between these two conditions there may arise situations in which during the course of solving a cooperative problem with the robot, the user comes to see that despite the "open endedness" of a given problem set, there may be repetitive subtasks that occur in a larger context in which some uncertainty can exist. In this type of situation, the human user may want to teach the robot about the repetitive part so this can be executed as an autonomous "macro" while the user still remains in the execution loop for the components that require his/her decision.

The table assembly task corresponds to this situation. For each of the four legs the robot should receive the leg from User2, pass it to User1 and then hold the table surface in place while User1 attaches the leg to the table, before repeating the same procedure for the next leg. After 1 or two repetitions of this exercise, for the first leg or two, User1 should have a good idea of how this repeating subsequence goes, and can thus teach it to the robot so that the entire behavior can be accessed by a single command.

III. IMPLEMENTATION

Based on the requirements derived from this scenario, we can now begin to allocate these requirements to different components of the system. The current studies are performed with the Kawada Industries HRP-2 humanoid robot [15] under the control of the OpenHRP controller [16]. The HRP-2 has 30 controlled degrees of freedom, 8 of which are used in this study. The spoken language interface technology is provided by the CSLU RAD system. This runs on a PC Pentium III Windows machine, which communicates with the OpenHRP controller via wireless internet with an ssh connection. The system is quite modular however, and the robot controller for the OpenHRP can be replaced by the controller for other robots. We have used the AIBO ERS7 with a WIFI interface, the Lynxmotion 6DOF robot arm, and Khepera mobile robots with a serial port controller [3]. Part of the novelty here is the use of the HRP-2 with many more effective degrees of freedom and possibilities for rich cooperative interaction.

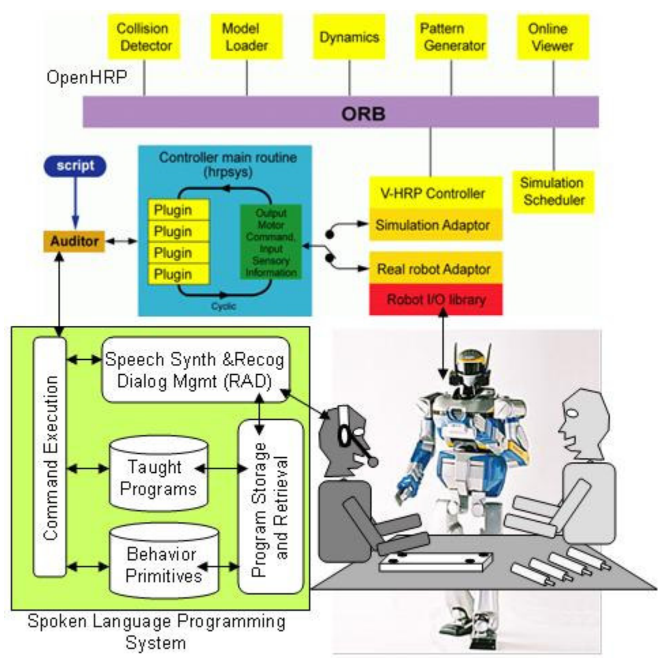

Fig. 1. Implementation architecture and stylized view of the cooperative interaction scenario. The two human users and the HRP-2 will cooperate in constructing the table.

A. Dialog Management

Dialog management and spoken language processing (voice recognition, and synthesis) is provided by the CSLU Rapid Application Development (RAD) Toolkit (http://cslu.cse.ogi.edu/ toolkit/). RAD provides a state-based dialog system capability, in which the passage from one state to another occurs as a function of recognition of spoken words or phrases; or evaluation of Boolean expressions.

In the mixed initiative dialog system we developed, the system prompts the user with "I am ready" and waits for the user to respond with one of the commands (Table 1) and these are immediately executed. The user can also issue commands for programming the robot (Table 2). These commands include "learn" and "ok" which indicate the beginning and end of a macro sequence to be stored. Thus, in a single session, a user might first operate in direct mode to become familiar with how to solve a given problem, then pass into macro learning mode, generate a new program and run it in order to simplify subsequent task execution.

B. *HRP-2 Specific Commands*

The behavioral result of a spoken action command that is issued either directly, or as part of a learned plan is the execution of the corresponding action on the robot. Based on the preliminary analysis of the table-building scenario described above, a set of primitive actions was identified for the HRP2. Each of these actions, specified in Table 1, corresponds to a particular posture or posture sequence that is specified as the angles for a subset of the 30 DOFs. These actions have been implemented in python scripts that specify final joint angles and motion durations for the given postures. The only existing HRP-2 capability we use is that of commanding joint angles and movement time in python scripts. Script execution is triggered remotely by the CSLU toolkit, and communicates directly with the low-level OpenHRP framework (Fig. 1). The motion is achieved by linearly interpolating joint angles between the starting and final configurations, for each specific action. We have chosen these simple actions in order to demonstrate the feasibility of the overall approach in the table-building scenario, with the expectation that they will generalize for application to other related tasks. More complex functions are currently under development.

Table 1. HRP-2 Specific Action Commands

Motor Command	Resulting Actions
Prepare	Move both arms to neutral position, rotate chest to center, elevate left arm, avoiding contact with the work surface (5 DOF)
OpenLeft	Open left hand (1 DOF)
CloseLeft	Close left hand (1 DOF)
Give it to me	Rotate hip to pass the object in left hand to User1, left in Fig 1 (1 DOF)
Hold	Center hip, raise right arm preparing to hold table top (5 DOF)
Right open	Open right hand (1 DOF)
Right close	Close right hand (1 DOF)

C. *General learning and control commands*

In addition to the HRP-2 specific motion commands, the system requires a set of commands that allow the user to control the actual programming and program execution. These commands and their consequences are presented in Table 2. When the user invokes the "Learn" command, the dialog system begins to encode the sequence of the subsequent commands that are issued. The user proceeds to issue action commands to effect the desired task that will make up this sequence. When the user has finished the part of the task he wants to program, he issues the "OK" command. This results in the action sequence being written to a file. Now, when the "Macro" command is issued, this file is read into an array, and the commands are sequentially executed. During these executions, the behavioral scenarios above also identified the requirement for a conditional wait, in which the execution of a stored sequence waits for the user to finish what he is doing which the user signifies with the "continue" command. Thus, when the "wait" condition is issued, the system pauses until the "continue" command is issued.

Table 2. SLP Learning and Control Commands

Commands	Correspondence
Learn	Begin encoding subsequent commands
OK	Store encoded command sequence in macro
Macro	Execute the stored macro
Wait	Interrupt command execution until a spoken "continue" command is issued
Continue	Terminate the "wait" pause and resume execution.

IV. Experimental Results

In order to evaluate the implemented system, we performed two experiments that involved human-robot cooperation via SLP. In the first experiment, two human users and the robot cooperate to construct a small table. In the second experiment, they cooperate again, this time to take the table apart. In our initial development, only the first experiment was planned, but when it was successfully accomplished we wanted to determine if the set of action primitives we had developed were indeed general enough to extend to a new but related task of disassembling the table.

Each of the two experiments will be presented, followed by an analysis of the effects of the ability to program on measures of efficiency of task completion.

A. *Experiment 1: Assembling the table*

In the following paragraphs we provide the literal transcription of the human-robot dialog, annotated with explanation about the corresponding human and robot behavior and key system functions in <> brackets. Robot language will be indicated in *italics*, and human language in plain text, with main commands in **bold.**

1) Phase 1. Spoken language commanding.

The users begin their first interaction with the robot. User1 controls the task via spoken language. Because the task is new to him, he proceeds for the moment only by commanding the robot, with no programming.

Dialog: <Robot initiates the interaction> *I am ready*. **Prepare**. *You said prepare?* Yes. <Because of possible errors in speech recognition, we choose to have an explicit validation on each command as illustrated here>. *Preparing to grasp.*<Robot raises left hand > *I am ready.* **Left open.** *You said left open?* Yes. *Opening left hand.* <Robot opens left hand> *I am ready.* <User2 places the table leg in the robot's left hand as in Fig2A)> **Left close**. *You said left close*. Yes. *Closing left hand.* <Robot closes left hand to grasp the table leg> *I am ready*. **Give it to me**. *You said give it to me?* Yes. *Giving you the object.* <Robot turns to its right, holding the table leg out to User1> *I am ready.*

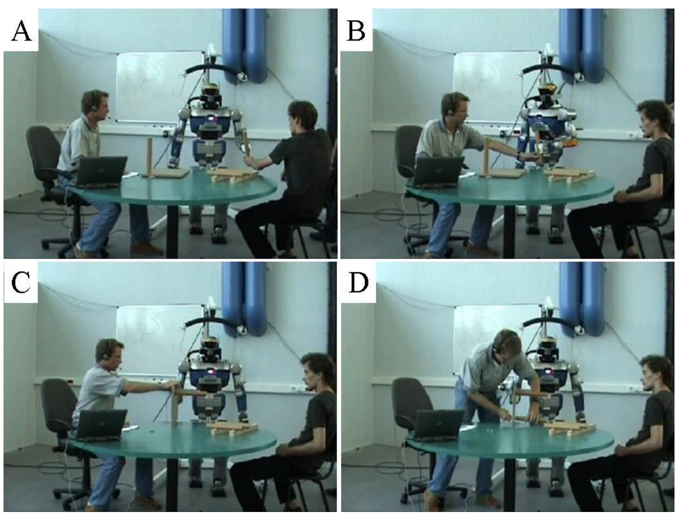

Figure 2. Cooperative task execution. A. User2 hands leg to Robot. B. Robt hands leg to User1. C. User1 gives table to Robot. D. Robot holds table while User1 attaches leg.

Left open. *You said left open?* Yes. *Opening left hand.* <As the Robot opens the hand, User1 reaches for and grasps the table leg (as in Fig. 2B)> *I am ready.* **Hold**. <the first of several primitives required for the robot to hold the table while User1 attaches the leg.> *You said give it to me?* No. <This is an example of a speech recognition error.> *I am ready.* Hold. *You said hold?* Yes. *Getting ready to hold.* <The robot orients to the work surface and raises its right hand. *I am ready*. **Right open**. *You said right open?* Yes. *Opening right hand.* <With the robot's right hand open, User1 places the table top within the robot's grasp (Fig. 2C). *I am ready*. **Right close**. *You said right close?* Yes. *Closing right hand. I am ready.* **Wait**. *You said wait?* Yes. *Waiting for your signal* <During this wait period, User1 attaches the leg to the table as in Fig2D, and then gives the "continue" command when ready.> **Continue**. *I am ready.* **Right open.** *You said right open?* Yes. *Opening right hand.* <User1 takes the table from the robot and prepares for the next leg.> *I am ready.*

2) Phase 2: Programming the Robot

At this point, User1 has used spoken language tele-operation to enlist the robot's assistance in attaching the first leg to the table, as indicated in Fig. 2. Given the knowledge of the task User1 has gained, he is now ready to "program" the robot for the repetitive component of the task by using the "learn" command.

Dialog: **Learn**. *You said learn?* Yes. *I'll start learning now. I am ready.* **Prepare**. *...The interaction continues as in Phase 1 until the end ...* **Continue**. *I am ready.* **Right open**. *You said right open?* Yes. *Opening right hand.* <User1 takes the table>. *I am ready.* OK. *You said OK?* Yes. *OK we will store this plan. I am ready.*

At this point the sequence of commands including the conditional wait are now stored as an SLP macro for immediate re-use.

3) First Execution of Stored Program Macro

Dialog: **Macro**. You said macro? Yes. Running the macro. Preparing to grasp. Waiting for your signal. **Continue**. Closing left hand. <Robot takes leg from U2> Giving you the object. Opening left hand. <U1 takes leg>. Getting ready to hold. Waiting for your signal. **Continue**. Closing right hand. Waiting for your signal <User1 attaches the leg>. **Continue**. Opening right hand. In line macro finished. I am ready. <Third leg is now attached. User1 takes the table. Same procedure with macro for attaching fourth leg.>

4) Performance Analysis of SLP Effects

During the assembly of the first two legs, User1 issued 10 distinct commands for each of the two legs. Installing legs 1 and 2 took 3min 25sec and 3min 24sec. Once the program was learned, for legs 3 and 4, a single command initiated the program, and the user was only required to issue 3 "continue" commands in order to indicate to the robot that he was ready to proceed. Execution time was reduced to 2:11 and 2:33 respectively. The ability to execute a suite of primitive behaviors with a single command had a significant impact on the execution time for completion of the task. Fig 4 indicates the average execution times for the individual commands under the explicit command conditions without the programmed macro (CMD) at 25.14 sec/operation, and under macro program execution conditions (PRG) at 15.00 sec/operation. We performed a statistical analysis of the variance (ANOVA) in these completion times examining the effects of Repetition (i.e. first and second trial in either the CMD or PRG mode), and Programming condition (i.e. CMD vs PRG). Only the Programming condition had a significant effect on the completion times (ANOVA, Programming Effect: $F(1,6) = 109$, $p < 0.0001$). These results demonstrate that SLP not only eliminates the need to enumerate the successive commands, it also significantly reduces the task completion time.

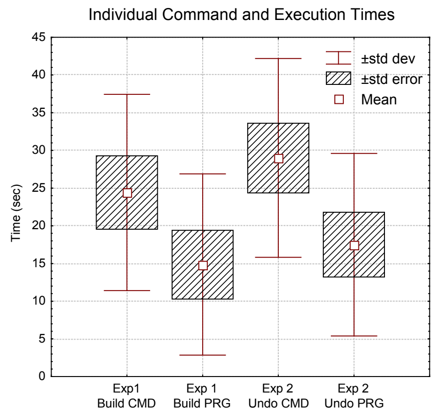

Fig. 3. Average command execution times for the Building (Exp1) and Undoing (Exp2) task using spoken language for on-line commanding (CMD) and for macro programming (PRG).

B. Experiment 2: Disassembling the Table

Part of the utility of a spoken language capability for robot programming (as for traditional programming) will be to allow the human user to program arbitrary behaviors, rather than just a single behavior. Experiment 2 thus tests whether the system that was developed for assembling the table could equally well serve to allow the user to program the robot to aid in disassembling the table.

Disassembly resembles the reverse process of assembly. User1 will first ask the robot to hold the table while he removes the leg. User1 will then put the robot into the "give it to me" posture, with the left hand open, give it the leg, and then issue to "prepare" command so that the robot passes the leg to User2 who then puts it away. For ease of reading, we dispense with the portion of the dialog in which the robot asks for confirmation of each command.

1) Phase 1. Spoken language commanding.

Dialog: *I am ready.* **Hold**. *Getting ready to hold. I am ready.* **Right open**. *Opening right hand. I am ready.* <User2 puts table in robot's hand> **Right close**. *Closing right hand.*<Robot grasps the table> *I am ready.* **Wait**. *Waiting for your signal.* <User1 unscrews the first leg while the robot holds the table> **Continue**. *I am ready.* **Right open**. *Opening right hand.* <Robot releases table, User1 takes the table and places it on the work surface> *I am ready.* **Prepare**. *Preparing to grasp.* <Right hand descends, left hand raises, preparing to take the leg> *I am ready.* **Give it to me** *Giving you the object.* <This is a bit counter-intuitive. With this command robot rotates its hip to face User1>. *I am ready.* **Left open.** *Opening left hand. I am ready.* <User1 places the first table leg within the robot's left hand.> **Left close**. *Closing left hand.* <Robot grasps the leg> *I am ready.* **Prepare**. *Preparing to grasp* <Robot orients to User2> . *I am ready.* **Left open.** <User2 takes the leg.> *I am ready.*

2) Phase 2: Programming the Robot

As in the previous experiment, after one run with the first leg, User1 is now ready to program the robot. Again, the user initiates the program storage by saying "Learn". He then executes step-by-step the procedure for taking a leg off and passing it to User2 with the help of the robot, and finally storing this program by saying "OK". The important point is that by using exactly the same primitives but in a different sequence we were able to generate a new stored macro on the fly for a different, but related, task, thus demonstrating the generalization capability of the SLP system.

3) First Execution of Stored Program Macro

I am ready. **Macro**. *Running the macro. Getting ready to hold.* <User1 places the table in the robot's right hand> *Closing right hand. Waiting for your signal.* < User1 unscrews the leg and then tells the robot to continue>. **Continue**. *Opening right hand* <Robot releases table, user1 places it on table surface> *Preparing to grasp.* <Right hand descends, left hand raises, preparing to take the leg> *Giving you the object.* <Robot rotates hip to face User1>. *Closing left hand.* <Robot takes the leg from User1>*Preparing to grasp.* <Robot orients to User2> *Opening left hand* <Robot gives the leg to User2> The second execution of the macro for the final leg is identical, and the table is thus taken apart.

4) Performance analysis

As in Experiment 1, the use of the programming capability for the third and fourth leg (executed in 2:51 and 2:51 respectively) yielded significant reductions in execution time as compared with the first two legs (executed in 3:57 and 4:11 respectively). To compare performance in the two experiments we performed a 3 way ANOVA with the factors Experiment (Exp1 vs. Exp2), Programming vs simple voice Commanding (PRG vs CMD), and Repetition (First vs. second repetition in each condition). Figure 3 indicates that both for Exp1 and Exp2 the completion times were elevated for the CMD vs PRG conditions, i.e. action execution was slower when programming was not used. The ANOVA reveled that only the Programming effect was significant ($F(1,6) = 277$, $p < 0.0001$).

V. Discussion

Over the past several years we have experimented with spoken language control of different robot systems including the AIBO ERS-7, the Khepera mobile robot, and the Lynx-6 arm (see http://dominey.perso.cegetel.net/RobotDemos.htm for video demos) [3-8]. Part of our goal in these efforts has been to develop a generic system for commanding and programming robots that can be rapidly adapted to new robotic platforms.

1) Lessons learned

In this context, the current research has yielded for the first time, the ability for a human user to employ spoken language to program a humanoid robot in real time to participate with humans in two distinct cooperative and complex object manipulation tasks. Despite this positive outcome, however, we have not yet fully exploited the potential richness of the predicate-argument structure of grammatical constructions. There are two important considerations to note here. First, a 3 month field study with an interacting robot [17] concluded that expectations on language-based interfaces have often been too high, and that "we need rich empirical experience of the use of robust and simple systems in order to formulate new and relevant questions," justifying our simplified (and successful) approach. Second, this simplified approach has aided us in generating requirements for higher level predicate argument representations for robot action and perception that will allow us to more deeply exploit the communicative richness of natural spoken language.

2) Related Research

Communicative interaction that will allow humans to truly cooperate with robots is an open and active area of research. Progress towards this objective is being made in part via well-documented methods for action learning that include demonstration and imitation [13-14]. Language has been used in this context for correcting and clarifying what is being learned by demonstration [9]. One of the fundamental requirements is to establish the grounded meaning at the base of the communication, that is the link between human language, and robot action and perception. This has recently been explored and developed in the domain of language based navigation [10, 11]. Roy and colleagues further establish these links via an amodal Grounded Situation Model that integrates perception, action and language in a common framework for language based human-robot cooperation [12]. We have made progress with a system that can learn grammatical constructions which make the mapping between predicate-argument representation of action as perceived by a robot vision system, and natural language sentences that describe that action, generalizing to new action scenes [3, 4]. In this context of language-based human-robot cooperation, the current research demonstrates - for the first time - the capability for a human user to tell a humanoid what to do in a cooperative task so that in real time, the robot performs the task, and acquires new skills that significantly facilitate the ongoing cooperative human-robot interaction.

ACKNOWLEDGMENT

The authors thank Jean-Paul Laumond, Co-Director of the French-Japanese Joint Robotics Laboratory of the CNRS for support, and comments on the ms. We also thank Hajime Saito of General Robotix for invaluable assistance in technical aspects of the interface to OpenHRP.

REFERENCES

[1] Crangle C. & Suppes P. (1994) Language and Learning for Robots, CSLI lecture notes: no. 41, Stanford.

[2] Goldberg A. Constructions: A new theoretical approach to language. Trends in Cognitive Sciences 2003; 7: 219–24.

[3] Dominey PF (2007) Spoken Language and Vision for Adaptive Human-Robot Cooperation, in (Matthias Hackel, Ed). *Humanoid Robotics*, ARS International, Vienna.

[4] Dominey PF, Boucher (2005) Learning To Talk About Events From Narrated Video in the Construction Grammar Framework, *Artificial Intelligence*, 167 (2005) 31–61

[5] Dominey, P.F., 2003. Learning grammatical constructions from narrated video events for human–robot interaction. Proceedings IEEE Humanoid Robotics Conference, Karlsruhe, Germany

[6] Dominey, P. F., Boucher, J. D., & Inui, T. (2004). Building an adaptive spoken language interface for perceptually grounded human–robot interaction. In Proceedings of the IEEE-RAS/RSJ international conference on humanoid robots.

[7] Dominey PF, Alvarez M, Gao B, Jeambrun M, Weitzenfeld A, Medrano A (2005) Robot Command, Interrogation and Teaching via Social Interaction, Proc. IEEE Conf. On Humanoid Robotics 2005

[8] Boucher J-D, Dominey PF (2006) Programming by Cooperation: Perceptual-Motor Sequence Learning via Human-Robot Interaction, Proc. Simulation of Adaptive Behavior, Rome 2006.

[9] Nicolescu M.N., Mataric M.J. : Learning and Interacting in Human-Robot Domains, IEEE Trans. Sys. Man Cybernetics B, 31(5) 419-430.

[10] Lauria S, Buggmann G, Kyriacou T, Klein E (2002) Mobile robot programming using natural language. Robotics and Autonomous Systems 38(3-4) 171-181

[11] Kyriacou T, Bugmann G, Lauria S (2005) Vision-based urban navigation procedures for verbally instructed robots. Robotics and Autonomous Systems, (51) 69-80

[12] Mavridis N, Roy D (2006). Grounded Situation Models for Robots: Where Words and Percepts Meet. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

[13] Zöllner R., Asfour T., Dillman R.: Programming by Demonstration: Dual-Arm Manipulation Tasks for Humanoid Robots. Proc IEEE/RSJ Intern. Conf on Intelligent Robots and systems (IROS 2004).

[14] Calinon S, Guenter F, Billard A (2006) On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts. Proc IEEE/ICRA 2006.

[15] K.Kaneko, F.Kanehiro, S.Kajita, H.Hirukawa, T.Kawasaki, M.Hirata, K.Akachi, and T.Isozumi, "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, vol. 2, 2004, pp. 1083–1090.

[16] F. Kanehiro, N. Miyata, S. Kajita, K. Fujiwara, H. Hirukawa, Y. Nakamura, K. Yamane, I. Kohara, Y. Kawamura, and Y. Sankai, "Virtual Humanoid Robot Platform to Develop Controllers of Real Humanoid Robots without Porting," Proc. Int. Conference on Intelligent Robots and Systems, pp. 1093-1099, 2001.

[17] Severinson-Eklund K., Green A., Hüttenrauch H., Social and collaborative aspects of interaction with a service robot, Robotics and Autonomous Systems 42 (2003) 223–234