11

Spoken Language and Vision for Adaptive Human-Robot Cooperation

Peter Ford Dominey
CNRS – University of Lyon
France

1. Introduction

In order for humans and robots to cooperate in an effective manner, it must be possible for them to communicate. Spoken language is an obvious candidate for providing a means of communication. In previous research, we developed an integrated platform that combined visual scene interpretation with speech processing to provide input to a language learning model. The system was demonstrated to learn a rich set of sentence-meaning mappings that could allow it to construct the appropriate meanings for new sentences in a generalization task. We subsequently extended the system not only to understand what it hears, but also to describe what it sees and to interact with the human user. This is a natural extension of the knowledge of sentence-to-meaning mappings that is now applied in the inverse scene-to-sentence sense (Dominey & Boucher 2005). The current chapter extends this work to analyse how the spoken language can be used by human users to communicate with a Khepera navigator, a Lynxmotion 6DOF manipulator arm, and the Kawada Industries HRP-2 Humanoid, to program the robots' behavior in cooperative tasks, such as working together to perform an object transportation task, or to assemble a piece of furniture. In this framework, a system for Spoken Language Programming (SLP) is presented. The objectives of the system are to 1. Allow the human to impart knowledge of how to accomplish a cooperative task to the robot, in the form of a sensory-motor action plan. 2. To allow the user to test and modify the learned plans. 3. To do this in a semi-natural and real-time manner using spoken language and visual observation/demonstration. 4. When possible, to exploit knowledge from studies of cognitive development in making implementation choices. With respect to cognitive development, in addition to the construction grammar model, we also exploit the concept of "shared intentions" from developmental cognition as goal-directed action plans that will be shared by the human and robot during cooperative activities.

Results from several experiments with the SLP system employed on the different platforms are presented. 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. Finally, in addition to language, we investingate how vision can be used by the robot as well to observe human activity in order to able to take part in the observed activities. At the interface of cognitive development and robotics, the results are interesting in that they (1) provide concrete demonstration of how cognitive science can contribute to human-robot interaction fidelity, and (2) they suggest how robots might be used to experiment with theories on the implementation of cognition in the developing human.

Source: Humanoid Robots, New Developments, Book edited by: Armando Carlos de Pina Filho
ISBN 978-3-902613-02-8, pp.582, I-Tech, Vienna, Austria, June 2007

2. Language and Meaning

Crangle and Suppes (1994) 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." Spoken language provides a very rich and direct means of communication between cooperating humans (Pickering & Garrod 2004). Language essentially provides a vector for the transmission of meaning between agents, and should thus be well adapted for allowing humans to transmit meaning to robots. This raises technical issues of how to extract meaning from language.

Construction grammar (CxG) provides a linguistic formalism for achieving the required link from language to meaning (Goldberg 2003). Indeed, grammatical constructions define the direct mapping from sentences to meaning. Meaning of a sentence such as (1) 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 employ abstract argument "variables" that can take an open set of values.

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

We previously developed a system that generates PA representations (i.e. meanings) from video event sequences. When humans performed events and described what they were doing, the resulting <sentence, meaning> input pairs allowed a separate learning system to acquire a set of grammatical constructions defining the sentences. The resulting system could describe new events and answer questions with the resulting set of learned grammatical constructions (Dominey & Boucher 2005).

PA representations can be applied to commanding actions as well as describing them. Hence the CxG framework for mapping between sentences and their PA meaning can be applied to action commands as well. In either case, the richness of the language employed will be constrained by the richness of the perceptual and action PA representations of the target robot system. In the current research we examine how simple commands and grammatical constructions can be used for action command in HRI. Part of the challenge is to define an intermediate layer of language-commanded robot actions that are well adapted to a class of HRI cooperation tasks.

This is similar to the language-based task analysis in Lauria et al. (2002). An essential part of the analysis we perform concerns examining a given task scenario and determining the set of action/command primitives that satisfy two requirements. 1. They should allow a logical decomposition of the set of tasks 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 different tasks can be performed with the same set of primitives.

3. Spoken Language Programming

For some tasks (e.g. navigating with a map) the sequence of human and robot actions required to achieve the task can be easily determined before beginning execution. Other tasks may require active exploration of the task space during execution in order to find a solution. In the first case, the user can tell the robot what to do before beginning execution, while in the second, instruction will take place during the actual execution of the task.

In this context, (Lauria et al. 2002) asked naïve subjects to provide verbal instructions to a robot in a miniature-town navigation task. Based on an analysis of the resulting speech corpora, they identified a set of verbal action chunks that could map onto robot control primitives. More recently, they demonstrated the effectiveness of navigation instructions translated into these primitive procedures for actual robot navigation (Kyriacou et al. 2005). This research indicates the importance of implementing the mapping between language and behavioural primitives for high-level natural language instruction or programming. The current study extends these results by introducing a conditional (e.g. if-then) component to the programming.

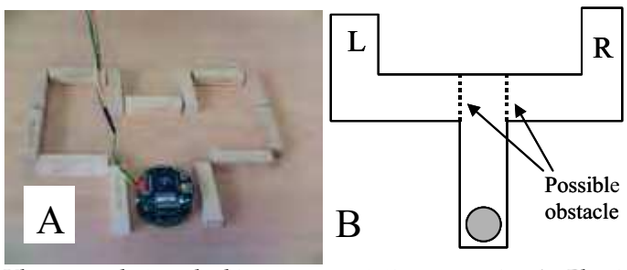

Fig. 1. Khepera robot and object transportation scenario. A. Physical set up. B. Labeled schematic representation.

Figure 1 illustrates a given scenario for HRI. In Figure 1A we see the Khepera robot on a table top, and in Figure 1B a schematic of the robot (represented as the gray disc) in a two arm maze. Consider a task in which User1 sends the Khepera robot to User2 who gives it an object to transport back to User1. Getting to User2 requires a conditional choice between two paths based on the location of an obstacle that is not known in advance, at the locations indicated by the dotted lines. Once the robot determines the location of the obstacle, it should choose the path that is not blocked, and make its way to the end of the arm. There it should turn around, and wait for User2, to place an object on its carrying surface. When User2 has performed this, User1 will say "continue" and the robot should then return to the beginning of the maze arm where User1 will take the transported block, and the process then continues.

On-Line Commanding: The simplest solution for controlling the robot in this task, which involves no learning, is for User1 simply to tell the robot what to do, step by step. Depending on whether the obstacle is at the left or right arm position, the human User1 would decide whether to tell the robot to take the right or left pathway. Likewise, once at the end point, User1 would wait until the User2 placed the object before commanding the robot to turn around and come back.

Programming a Pre-Specified Problem: Again, in many cases, the problem is known to the user in advance (possibly because the user has now "walked" the robot through the problem as just described) and can be specified prior to execution.

The objective of SLP is to provide a framework in which non-specialist humans can explain such a task to a robot, using simple spoken language. While the task described above remains relatively simple, explaining it to the robot already levies several interesting requirements on the system, and meeting these requirements will provide a fairly general and robust capability for SLP. In particular, this task requires

the following: (1) The user should be able to provide a sequence of primitive commands that will be executed in a particular order. (2) The user should be able to specify conditional execution, based on the values of robot sensors. (3) The user should be able to tell the robot that at a certain point it should wait for some sensory event to occur before proceeding with its sequence execution., as demonstrated below.

In addition, for tasks that become increasingly long and complex, the user may make mistakes in his/her initial specification of the task, so the SLP system should allow the user to test, "debug" and modify programs. That is, once the user has specified the program, he should be able to execute it and – if there is a problem – modify it in a simple and straightforward manner.

On-line Commanding with Repetitive Sub-Tasks: 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 can program the robot as defined above. 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 repetitive subtasks that occur in a larger context. 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 subroutine or "macro" while the user still remains in the execution loop for the components that require his/her decision.

Fig. 2. Human-Robot cooperation in a furniture construction task. A. Robot takes a table leg from User2. B. Robot hands the leg to User1. C. Robot holds the table steady while User1 attaches the leg.

Figure 2 illustrates such an HRI scenario that involves two humans and the HRP-2 cooperating in the construction of a small table. The construction task will involve fixing 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 fixed to the table, and the "get the leg and attach it" sequence can be repeated. This task thus has a repeating subsequence that can be applied to each of the four legs of the table. Experiments below address this task. These experiments identify the utility of a more complex command structure based on grammatical constructions.

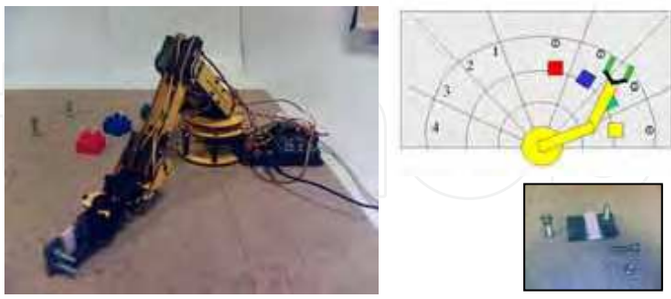

Fig. 3. Cooperative manipulation task scenario. User asks for screws from locations identified by color. While robot holds the plastic "table" (as illustrated) the user can insert the screws into the table (see inset).

Influence of Robot Command Structure on Language: Part of the central principal of construction grammar (and the related domain of cognitive linguistics) is that human language is made up of a structured inventory of grammatical constructions (Goldberg 2003). These constructions reflect the meaning of prototypical events that are basic to human experience. Thus for example the "ditransitive" construction involves one agent transferring an object to another, as in "John gave Mary the ball." From a robotics perspective, the action predicate-argument Move(Object, Location) can thus be considered as a template for a motor program that allows the use of constructions of the form "Give the X to Y". This robot PA involves localizing X and Y, and then grasping X and transporting the effector to Y to transfer X to Y. To the extent that such an action PA (APA) is built into the repertoire of the robot (or to the extent that it can be built up from primitives like localize(X), transport-to(X) etc.) rich grammatical constructions can be used to generate and interpret sentences like "give me the red block," or "put the blue block next to the red block" for robot control.
Figure 3 illustrates a simplified version of the table construction scenario that uses the Lynx6 arm for cooperative manipulation in a construction task. Experiments below examine cooperative construction with the Lynx6 based on lessons learned from the HRP-2 experiments, including more complex commands based on grammatical constructions.

4. Implementation of Spoken Language Programming

Because of the potential influence that the robot itself will have on spoken language interaction, we chose to develop a core SLP system, and then to adapt the system to three different robot platforms the K-Team Khepera, the LynxMotion Lynx 6 DOF manipulator arm, and the Kawada Industries HRP-2 Humanoid. This allows us to explore the navigation vs. manipulation dimension, and within the manipulation dimension the Lynx allows for rapid prototyping while the HRP-2 allows for a much more robust and human-scale human-robot interaction.
Spoken Language Programming Core: The core SLP system is presented in Figures 4 and 5, and Table 1. The core system provides the command, conditional control, and programming/teaching capabilities based on spoken language.

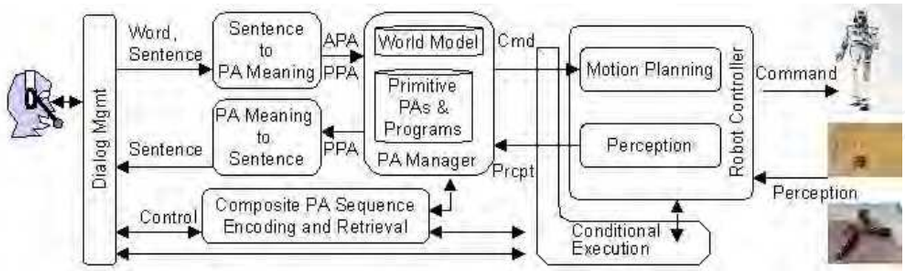

Fig. 4. Spoken Language Programming Architecture. The fundamental unit of meaning is the predicate-argument (PA) representation. APA – Action Predicate-Argument representation. PPA – Perceptual Predicate-Argument representation. Spoken language commands are interpreted as individual words or grammatical constructions, and the command and possible arguments are extracted in Sentence to PA Meaning. The robot level commands (CMD) are extracted by the PA manager. World Model includes specification of object locations, and related state information. Robot commands are then issued to the Controller to effect the commands. Conditional commands including IF *CONDITION* .. OTHERWISE, and WAIT are implemented as part of the Conditional Execution. Learning-related commands LEARN, OK/SAVE-MACRO, RUN MACRO are handled by the Sequence Encoding and Retrieval function. Event perception and description, provided by PA Meaning to Sentence are implemented in Dominey & Boucher (2005).

Dialog management and spoken language processing (voice recognition, and synthesis) is provided by an "off the shelf" public domain product, the CSLU Rapid Appication 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. Figure 5 illustrates the major dialog states.

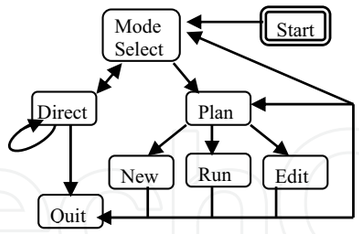

Fig. 5. Major Dialog States

In Figure 5, when the system starts in the "Mode Select" state it asks the user "*Shall we learn by planning or by demonstration?*"

Learning by Demonstration: If the user replies "demonstration" (note: In all dialogs, human responses will be indicated by *italics*) then the system transitions to the "Direct" state, and interrogates the subject with "*Now what?*" The user can reply with one of the robot-specific motor commands, which the robot will execute and then again prompt "*Now what?*" In the course of this direct interaction, if the user

determines that there is a repetitive component in the task, as illustrated in the discussion of Figure 2, above, the user can invoke the "Learn" command. This indicates to the system to begin to store the subsequent commands. When the repetitive subsequence has been performed, the user then issues the command "OK" and the subsequence is stored, and can then be executed in-line with the command "Macro". This learning, in which the user "demonstrates" the performance by directing the robot through the execution is illustrated in Exps 3-5.

Learning by Planning: In Figure 5, when the system starts in the "Mode Select" state it asks the user "*Shall we learn by planning or by demonstration?*" If the user replies "planning" then the system transitions to the "Plan" state, and interrogates the subject with "*Shall we make a new plan, run the old plan, edit it or quit ?*"

If the user replies "New plan" then the system continuously interrogates the user with "*now what?*," with the user responding each time with the next motor command or conditional command until the user replies "stop." In this manner, command by command without execution, the user specifies the planned program using spoken language. After "stop", the plan is stored, and the system returns to the Plan state. Here, the user can now run the plan to see how it works. If the user detects that there is a problem with one of the elements in the stored plan, at the "Plan" state, he has the option of saying that he wants to "edit" the plan.

Editing Learned Programs: When the system enters the Edit state, it displays the plan graphically as a list of commands and corresponding conditions, and then ask the user what he wants to modify. The user can then respond with a two word phrase in which the first word is "action" or "condition", and the second word is the number of the action or condition in question. The system then asks the user for the new value for that component, and then again asks what the user wants to modify. When the editing is finished, the system then returns to the Plan state where the user can either execute the modified plan, or do something else.

Conditional Execution: In addition to the learning-related commands, we have also identified the requirement for commands related to conditional behavior execution and control. These commands are specified in Table 1. Perhaps the most obvious of these the "if –then-else" construction.

Conditional Commands	**Effects / Correspondence**
If *condition*	Begin a conditional segment
Otherwise	End a conditional segment
Wait *condition*	Interrupt execution until *condition* is met
Continue	User command to proceed when waiting This is one of the conditions for wait.
Learning-Related Commands	
Learn	Begin macro learning
Ok/Stop	End macro learning
Macro	Run stored macro

Table 1. Conditional and Learning-Related Commands.

Indeed this is the first concrete example of the use of a multi-word construction. When the "if *condition*" construciton is issued, the user specifies the "if" and the corresponding condition in the same utternence (e.g. "if left clear"), and the "if" and the "left clear"

condition are paired together in the stored sequence plan. The user then continues to specify the succession of commands that should logically follow in case the "if" condition is satisfied. When the user terminates that logical section, he indicates this with the conditional command "otherwise". The subsequent succession of commands corresponds to those that should be executed in case the condition of the "if" fails. During sequence execution in the Run state, when the system encounters an "if", it tests the corresponding condition. When the condition is true, the system executes up to the "otherwise" statement. When false, it skips to the otherwise, and then resumes execution of the subsequent commands.

The behavioral scenarios above also identified the need for a conditional wait, in which the execution of a stored sequence waits for a sensory condition to become true. Like the "if", the "wait *condition*" construction uses the identifier "wait" followed by a condition, which is paired with the wait command. Then, during execution of the plan in the Run state, when the system encounters a "wait" action, it tests the corresponding condition, and waits until that condition is true before proceeding. The "continue" condition (indicated by the user saying "continue") is the single robot-independent condition.

Robot Specific Command: Given the robot-independent component of the SLP architecture, we now explain how this is adapted to the specific robots. The final behavioral result of a robot-specific action command that is issued either directly or as part of a learned plan is the execution of the corresponding action on the robot. We new specify the specific commands for each of the three robot platforms we used.

K-Team Khepera: The Khepera (K-Team) is a small two-wheeled robot equipped with proximity sensors that has been extensively explored in sensory-motor robotics. Based on the requirements for the cooperative task described above, a set of primitive actions, and senor states was identified for the Khepera. These robot-specific sensory-motor functions are identified in Table 2.

The Khepera is connected to the SLP system PC via the RS232 serial port. Commands are issued to the Khepera via this connection, and the values of the forward, left and right position sensors are read from this connection as well.

Motor Commands	**Resulting Actions**
Explore	Move forward until obstacle is encountered
Left	Turn 90° left
Right	Turn 90° right
Specifiable Conditions	**Corresponding check**
Left clear	Check left proximity sensor
Right clear	Check right proximity sensor
Obstacle removed	Check forward proximity sensor

Table 2. Khepera Action Commands and Sensory Conditions.

Kawada Industries HRP-2 Humanoid: Based on the preliminary analysis of the table-building scenario above, a set of primitive actions was identified for the Kawada Industries HRP-2 humanoid robot under the control of the OpenHRP controller (Dominey et al. 2007). The HRP-2 has 30 controlled degrees of freedom, 8 of which are used in this study. Each of the functions in Table 3 corresponds to a particular posture that is specified as the angles for a subset of the 30 DOFs. These actions have been implemented by hand-

written python scripts that specify final, hardcoded, joint angles and motion durations for the given postures. The python script execution is triggered remotely by the CSLU toolkit, and communicates directly with the low-level OpenHRP framework. 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, and more complex functions are currently under development.

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)
Left open	Open left hand (1 DOF)
Left close	Close left hand (1 DOF)
Give it to me	Rotate hip to pass the object in left hand to user on the right (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)

Table 3. HRP-2 Action Commands.

Lynx 6 DOF Manipulator: We also performed SLP experiments with a 6DOF Lynx 6 robot manipulator arm. The robot controller is connected to the Pentium PC running RAD via the RS232 serial port, and provides access to the 6 joint angles which can be controlled individually and in parallel with a time-to-completion parameter. Based on task requirements for object manipulation and the miniature task building tasks, we pre-localized 8 locations in the workspace along a fixed radius that correspond to the color and number locations indicated in Fig. 3. We then manually determined the joint angles for taking the manipulator to these regions, above the workspace, and for lowering the manipulator to be able to grasp objects at these locations. These postures were then used to create the action commands in Table 4.

Action Command	**Resulting Actions**
Put the *Color* screw at location *number*.	Move to location marked by *color*, pick up the screw, place it at location marked by *number*. Return to neutral position.
Give me the *color* screw.	Move to location marked by *color*, pick up the screw, place it at location three. Return to neutral position.
Open gripper	Open gripper
Close gripper	Close gripper
Hold the table	Move gripper to location 3, preparing to hold the table

Table 4. Lynx 6 Action Commands

Note that in Table 4 that the "put" and "give me" commands are multiword constructions that take arguments. These constructions correspond to the ditransitive construction from section II.B.

5. Experimental Results Part 1

We now provide results from a series of experiments that were performed based on the scenarios described in Section 3. These experiments were performed with human users who were familiar with the system.

5.1 Programming Conditional Navigation with Khepera

The current experiment tests whether the user can use spoken language to explain a complex conditional task to the robot, based on the scenario described in section 3 and Fig1 in which User1 sends the Khepera robot to User2 who gives it an object to transport back to User1. Getting to User2 requires a conditional choice between two paths based on the location of an obstacle that is not known in advance.
In the following dialog, robot language is specified in *italics*, and human responses in normal text. Technical and explanatory annotation is in <>'s.

Dialog for Cooperative Navigation and Transportation:
Shall we learn by planning or by demonstration ? Planning. *Shall we make a new plan, run the old plan, edit it or quit ?* New plan. *Now what? <user begins to specify the plan>*
Explore. *Now what?* If left clear. <the user specifies that if the left is clear, the following should be performed >
Now what? Left. *Now what?* Explore *Now what?* Right *Now what?* Explore. *Now what?* Left. *Now what?* Left. <the user has now specified the plan segment that takes the robot to location L in Fig 1, and turns it around, ready for the return trip>
Now what? Wait until I say continue. <user specifies that before continuing execution of the return trip the robot should wait for a verbal "continue" signal. During this period the second human team member places the Lego block on the robot. Speech pattern match on "wait *any continue" where *any matches anything.>
<The following plan segment takes the robot back to the starting location, where the principal team member can decide where to place the block.>
Now what? Explore. *Now what?* Left. *Now what?* Explore. *Now what?* Right. *Now what?* Explore. *Now what?* Left? *Now what?* Left. *Now what?*
< The planning for the conditional execution of the path to the left is finished. This plan segment now specifies the "else' - what should happen if the left arm is blocked. It is similar to what happens for the left arm case.>
Otherwise. *Now what?* Right. *Now what?* Explore. *Now what?* Left. *Now what?* Explore. *Now what?* Left. *Now what?* Left. *Now what?* Wait until I say continue. <Robot now at the R, ready to come back after User2 places the block.> *Now what?* Explore. *Now what?* Left. <This is an error that will turn the robot into the wall> *Now what?* Explore. *Now what?* Right. *Now what?* Explore. *Now what?* Left *Now what?* Left. *Now what?* Stop. <Stop indicates that the program should be stored, and then executed>

When the stored plan executed we observed that when the obstacle was placed to the right, the robot correctly detected that the left was clear, and proceeded along the left arm to the location marked L. It then announced that a conditional wait was encountered, and that it was waiting for the spoken command "continue". This allowed User1 to verify that User2 had placed a Lego on the robot, so that he could then say "continue" and the robot continued back to the starting location, where User1 could recover the block. We then placed the obstacle at the left location. The conditional execution of the "if" determined that

the test condition failed, and execution skipped to the "otherwise" section of the plan. In execution of this plan segment, an error in the plan was detected. After the robot started to return to User1 (after the "continue" command had been issued) the plan incorrectly specified that the robot turn left (into the wall) rather than right (towards the center of the maze).
That is, during the specification of a plan of this length, it is possible that the user makes an error in a command, particularly in left vs. right distinctions when the robot is moving towards the human, in which the human and robot reference frames are reversed. The editing capability described above thus allowed the user to listen to the program and see it on the screen and then modify the errors by indicating the number of the action in the program, and what it should be changed to.

5.2 Programming Cooperative Manipulation with HRP-2

We now move from the navigation domain to object manipulation. The current experiment exploits the bimanual manipulation capabilities of the HRP-2 humanoid for a human-robot cooperation task that involves constructing a piece of furniture (Dominey et al. 2007).
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. Again, robot language will be indicated in *italics*, and human language in plain text.

Phase 1. Spoken language commanding.
Users 1 and 2 (Fig 3) 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 with the prompt> *I am ready*. <the human responsd with the command> 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. *Opening left hand.* <Robot opens left hand> *I am ready.* <User2 places the table leg in the robot's left hand (See Fig 2)> Left close. *Closing left hand.* <Robot closes left hand to grasp the table leg> *I am ready*. Give it to me. *Giving you the object.* <Robot turns to the right, holding the table leg out to User1> *I am ready.* Left open. Yes. *Opening left hand.* <As the Robot opens the hand, User1 reaches for and grasps the table leg> *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. Yes. *Getting ready to hold.* <The robot orients to the work surface and raises its right hand. *I am ready*. Right open. *Opening right hand.* <With the robot's right hand open, User1 places the table top within the robot's grasp>. *I am ready*. Right close. *Closing right hand. I am ready.* Wait. *Waiting for your signal* <During this wait period, User1 attaches the leg to the table, and then gives the "continue" command when ready.> Continue. *I am ready.* Right open. *Opening right hand.* <User1 takes the table from the robot and prepares for the next leg.> *I am ready.*

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. 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. *You said prepare?* Yes. *Preparing to grasp....*(same sequence as for the first leg) ... 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.*

Execution of Stored Program Macro
Now the stored program can be used via the "macro" command.
Dialog: Macro. *You said macro?* Yes. *Running the macro. Preparing to grasp. Waiting for your signal*. Continue. *Closing left hand.* <Robot takes leg from User2> *Giving you the object. Opening left hand.* <User1 takes leg>. *Getting ready to hold. Waiting for your signal.* <User1 place table in robot's right hand> Continue. *Closing right hand. Waiting for your signal* <User1 attaches the table 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.>
Performance Analysis: 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. We performed a statistical analysis of the variance (ANOVA) in indivisual action completion times examining the effects of Repetition (i.e. first and second leg 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: $F(1,6) = 109$, $p < 0.0001$).
We performed a second experiment in which the same primitives were used, with absolutely no change to the software, to disassemble the table. 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 (assembly, disassembly), Programming (with vs. without, i.e. PRG vs CMD), and Repetition (First vs. second repetition in each condition). For 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 ($F(1,6) = 277$, $p < 0.0001$).

5.3 Grammatical Constructions for Manipulation with Lynx6 Arm

The programming by command sequencing illustrated in the previous experiment clearly leads to improvements in performance and facilitation of the task for the human. Further progress can be made by the implementation of higher level action functions that respect the natural interaction categories that can arise in these cooperative manipulation tasks. In particular, the user who is building the table repeatedly interacts with the robot so that the robot will pass him the next leg of the table. The action semantics can be captured in the construction "Give me X", or more generally "Move the object at location X to location Y" where X specifies a target object location and Y a destination location where that object is to be placed.
We hypothesized that the availability of this kind of construction would simplify the task for the user, and also speed up the execution process. We test this hypothesis with the Lynx

6 robot, as a prototype platform before subsequent implementation on the HRP-2. The table construction task with the Lynx employed a plastic table surface and four 3cm threaded bolts with nuts (see insert Fig.3). These served as legs, and the robot was used to fetch the legs and hold the table while the user fixed the legs to the table, as in the previous experiment. As illustrated in Fig. 3 we designated 4 areas indicated by color for the four legs to be placed initially, and four numbered locations where these legs could be placed using a construction of the form: Put the *X* leg at location *Y*, where X was one of the four color designations and Y one of the four numbered designations (see Fig 3). We also implemented a construction of the form : Give me the *X* leg, which commanded the robot to fetch the leg at location color, and place it at location 3 which was just in front of the user.

Dialog*:*
Command an action. Give me the red screw. *You want the red screw?* Yes. Giving the red screw <robot takes screw at red location> to you <and passes it to the user>. *Command an action.* Start learning. <this initiates memory of the subsequent command sequence> *Command an action.* Hold. *Preparing to hold. Command an action.* Close gripper. *Closing gripper. Command an action.* Wait. *Waiting for your signal.* <User insertes and tightens screw while robot holds the grey plastic "table". See Fig 3> Continue. *Command an action.* Open. *Opening gripper. Command an action*. Save the macro. *You said save the macro?* Yes.
Command an action. Give me the blue screw. *You want the blue screw?* Yes. *Giving the blue screw… to you. Command an action.* Run the macro. *You said run the macro?* Yes. *Preparing to hold. Closing gripper. Waiting for your signal.* <User attaches screw while robot holds the table steady>. Continue. *Opening gripper.*
Command an action. Give me the green screw. *You want the green screw?* Yes. *Giving the green screw… to you. Command an action.* Run the macro. *You said run the macro?* Yes. *Preparing to hold. Closing gripper. Waiting for your signal.* Continue. *Opening gripper.*
Command an action. Give me the yellow screw. *You want the yellow screw?* Yes. *Giving the yellow screw… to you. Command an action.* Run the macro. *You said run the macro?* Yes. *Preparing to hold. Closing gripper. Waiting for your signal.* Continue. *Opening gripper.*
The total execution time for the assembly was less than five minutes. With the macro use, the user required five spoken commands per leg, and three of these were confirmations of commands and the continue signal. The two main commands are "Give me.." and "run the macro." With respect to the HRP-2, the availability of the "Give me the X screw" PA command eliminated the need to sequence a number of lower level postural commands into the macro, which instead was limited to implementing the "hold the table" function.

6. Programming Shared Behavior by Observation and Imitation

If spoken language is one of the richest ways to transmit knowledge about what to do, demonstration and imitation of observed behavior are clearly another. One of the current open challenges in cognitive computational neuroscience is to understand the neural basis of the human ability to observe and imitate action. The results from such an endeavor can then be implemented and tested in robotic systems. Recent results from human and non-human primate behavior, neuroanatomy and neurophysiology provide a rich set of observations that allow us to constrain the problem of how imitation is achieved. The current research identifies and exploits constraints in these three domains in order to develop a system for goal directed action perception and imitation.

One of the recurrent findings across studies of human imitation is that in the context of goal directed action, it is the goal itself that tends to take precedence in defining what is to be imitated, rather than the means (Bekkering et al. 2000, Tomasello et al. 2005). Of course in some situations it is the details (e.g. kinematics) of the movement itself that are to be imitated (see discussion inCarpetner and Call 2007, Cuijpers et al. 2006), but the current research focuses on goal based imitation. This body of research helped to formulate questions concerning what could be the neurophysiological substrates for goal based imitation. In 1992 di Pellegrino (et al.) in the Rizzolatti lab published the first results on "mirror" neurons, whose action potentials reflected both the production of specific goal-directed action, and the perception of the same action being carried by the experimenter. Since then, the premotor and parietal mirror system has been studied in detail in monkey (by single unit recording) and in man (by PET and fMRI) reviewed in Rizzolatti & Craighero (2004).

In the context of understanding imitation, the discovery of the mirror system had an immense theoretical impact, as it provided justification for a common code for action production and perception. In recent years a significant research activity has used simulation and robotic platforms to attempt to link imitation behavior to the underlying neurophysiology at different levels of detail (see Oztop et al. (2006) for a recent and thorough review, edited volume (Nehaniv & Dautenhahn 2007), and a dedicated special issue of Neural Networks (Billard & Schaal 2006)). Such research must directly address the question of how to determine what to imitate. Carpenter and Call (2007) distinguish three aspects of the demonstration to copy: the physical action, the resulting change in physical state, and the inferred goal – the internal representation of the desired state. Here we concentrate on imitation of the goal, with the advantage of eliminating the difficulties of mapping detailed movement trajectories across the actor and imitator (Cuijpers et al. 2006).

Part of the novelty of the current research is that it will explore imitation in the context of cooperative activity in which two agents act in a form of turn-taking sequence, with the actions of each one folding into an interleaved and coordinated intentional action plan. With respect to constraints derived from behavioral studies, we choose to examine child development studies, because such studies provide well-specified protocols that test behavior that is both relatively simple, and pertinent. The expectation is that a system that can account for this behavior should extend readily to more complex behavior, as demonstrated below.

Looking to the developmental data, Warneken, Chen and Tomasello (2006) engaged 18-24 month children goal-oriented tasks and social games which required cooperation. In one of the social games, the experiment began with a demonstration where one participant sent a wooden block sliding down an inclined tube and the other participant caught the block in a tin cup that made a rattling sound. This can be considered more generally as a task in which one participant moves an object so that the second participant can then in turn manipulate the object. This represents a minimal case of a coordinated action sequence. After the demonstration, in Trials 1 and 2 the experimenter sent the block down one of the tubes three times, and then switched to the other, and the child was required to choose the same tube as the partner. In Trials 3 and 4 during the game, the experimenter interrupted the behavior for 15 seconds and then resumed.

Behaviorally, children successfully participated in the game in Trials 1 and 2. In the interruption Trials 3 and 4 they displayed two particularly interesting types of response that were (a) to attempt to perform the role of the experimenter themselves, and/or (b) to

reengage the experimenter with a communicative act. This indicates that the children had a clear awareness both of their role and that of the adult in the shared coordinated activity. This research thus identifies a set of behavioral objectives for robot behavior in the perception and execution of cooperative intentional action. Such behavior could, however, be achieved in a number of possible architectures.

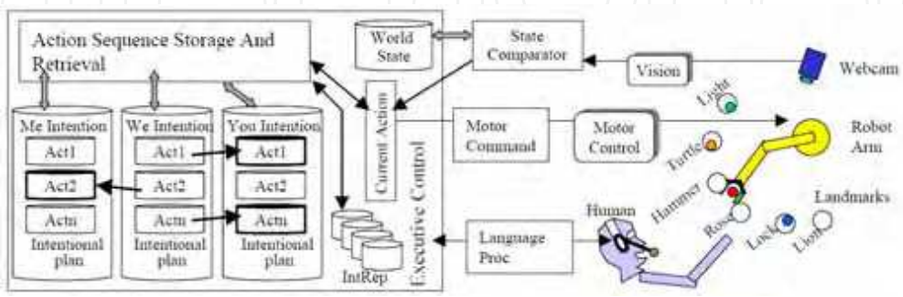

Fig. 6. Cooperation System. In a shared work-space, human and robot manipulate objects (green, yellow, read and blue circles corresponding to dog, horse, pig and duck), placing them next to the fixed landmarks (light, turtle, hammer, etc.). *Action*: Spoken commands interpreted as individual words or grammatical constructions, and the command and possible arguments are extracted using grammatical constructions in Language Proc. The resulting Action(Agent, Object, Recipient) representation is the Current Action. This is converted into robot command primitives (Motor Command) and joint angles (Motor Control) for the robot. *Perception*: Vision provides object location input, allowing action to be perceived as changes in World State (State Comparator). Resulting Current Action used for action description, imitation, and cooperative action sequences. *Imitation*: The user performed action is perceived and encoded in Current Action, which is then used to control the robot under the supervision of Executive Control. *Cooperative Games*. During observations, individual actions are perceived, and attributed to the agent or the other player (Me or You). The action sequence is stored in the We Intention structure, that can then be used to separately represent self vs. other actions.

7. Implementing Shared Planning

In a comment on Tomasello et al (2005) on understanding and sharing intention, Dominey (2005) analyses how a set of initial capabilities can be used to provide the basis for shared intentions. This includes capabilities to : 1. perceive the physical states of objects, 2. perceive (and perform) actions that change these states, 3. distinguish between self and other, 4. perceive emotional/evaluation responses in others, and 5. learn sequences of predicate-argument representations.

The goal is to demonstrate how these 5 properties can be implemented within the constraints of the neurophysiology data reviewed above in order to provide the basis for performing these cooperative tasks. In the current experiments the human and robot cooperate by moving physical objects to different positions in a shared work-space as illustrated in Figures 6 and 7. The 4 moveable objects are pieces of a wooden puzzle, representing a dog, a pig, a duck and a cow. These pieces can be moved by the robot and the user in the context of cooperative activity. Each has fixed to it a vertically protruding metal

screw, which provides an easy grasping target both for the robot and for humans. In addition there are 6 images that are fixed to the table and serve as landmarks for placing the moveable objects, and correspond to a light, a turtle, a hammer, a rose, a lock and a lion, as partially illustrated in Figures 6 & 7. In the interactions, human and robot are required to place objects in zones next to the different landmarks, so that the robot can more easily determine where objects are, and where to grasp them. Figure 6 provides an overview of the architecture, and Figure 7, which corresponds to Experiment 6 provides an overview of how the system operates.

Representation: The structure of the internal representations is a central factor determining how the system will function, and how it will generalize to new conditions. Based on the neurophysiology reviewed above, we use a common representation of action for both perception and production. Actions are identified by the agent, the object, and the target location to move that object to. As illustrated in Figure 6, by taking the short loop from vision, via Current Action Representation, to Motor Command, the system is thus configured for a form of goal-centered action imitation. This will be expanded upon below.

A central feature of the system is the World Model that represents the physical state of the world, and can be accessed and updated by vision, motor control, and language, similar to the Grounded Situation Model of Mavridis & Roy (2006). The World Model encodes the physical locations of objects that is updated by vision and proprioception (i.e. robot action updates World Model with new object location). Changes in the World Model in terms of an object being moved allows the system to detect actions in terms these object movements. Actions are represented in terms of the agent, the object and the goal of the action, in the form MOVE(object, goal location, agent). These representations can be used for commanding action, for describing recognized action, and thus for action imitation and narration, as seen below. In order to allow for more elaborate cooperative activity, the system must be able to store and retrieve actions in a sequential structure.

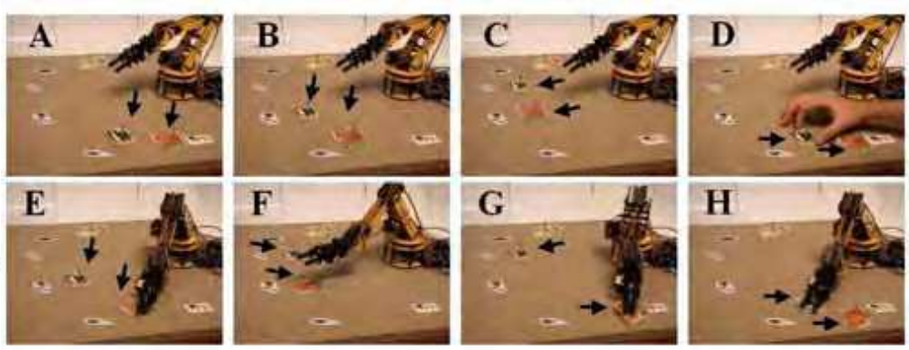

Fig. 7. Cooperative task of Exp 5-6. Robot arm, with 6 landmarks (Light, turtle, hammer, rose, lock and lion from top to bottom). Moveable objects include Dog and Horse. In A-D, human demonstrates a "horse chase the dog" game, and successively moves the Dog then Horse, indicating that in the game, the user then the robot are agents, respectively. After demonstration, human and robot "play the game". In each of E – F user moves Dog, and robot follows with Horse. In G robot moves horse, then in H robot detects that the user is having trouble and so "helps" the user with the final move of the dog. See Exp 5 & 6.

Visual perception: Visual perception is a challenging technical problem. To simplify, standard lighting conditions and a small set (n = 10) of visual object to recognize are employed (4 moveable objects and 6 location landmarks). A VGA webcam is positioned at 1.25 meters above the robot workspace. Vision processing is provided by the Spikenet Vision System (http://www.spikenet-technology.com/). Three recognition models for each object at different orientations (see Fig. 8.I) were built with an offline model builder. During real-time vision processing, the models are recognized, and their (x, y) location in camera coordinates are provided. Our vision post-processing eliminates spurious detections and returns the reliable (x, y) coordinates of each moveable object in a file. The nearest landmark is then calculated.

Motor Control & Visual-Motor Coordination: While visual-motor coordination is not the focus of the current work, it was necessary to provide some primitive functions to allow goal directed action. All of the robot actions, whether generated in a context of imitation, spoken command or cooperative interaction will be of the form *move(x to y)* where *x* is a member of a set of visually perceivable objects, and *y* is a member of the set of fixed locations on the work plan.

Robot motor control for transport and object manipulation with a two finger gripper is provided by the 6DOF Lynx6 arm (www.lynxmotion.com). The 6 motors of the arm are coordinated by a parallel controller connected to a PC computer that provides transmission of robot commands over the RS232 serial port.

Human users (and the robot) are constrained when they move an object, to place it in one of the zones designated next to each of the six landmarks (see Fig 3). This way, when the nearest landmark for an object has been determined, this is sufficient for the robot to grasp that object at the prespecified zone.

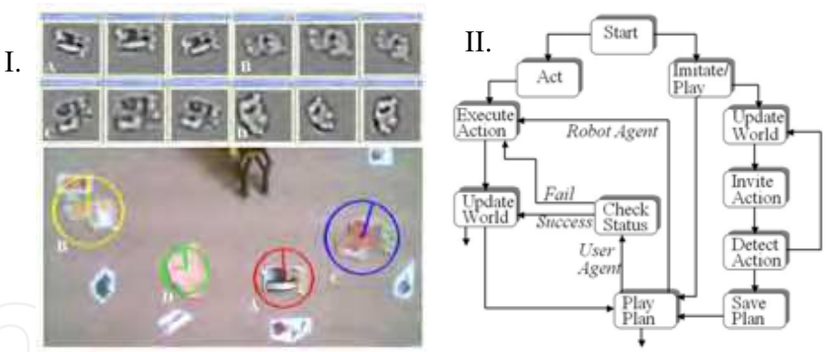

Figure 8. I. Vision processing. Above: A. – D. Three templates each for the Dog, Duck, Horse and Pig objects at three different orientations. Below, encompassing circles indicate template recognition for the four different objects near different fixed landmarks, as seen from the camera over the robot workspace. II. Dialog flow of Control

In a calibration phase, a target point is marked next to each of the 6 fixed landmark locations, such that they are all on an arc that is equidistant to the center of rotation of the robot arm base. For each, the rotation angle of Joint 0 (the rotating shoulder base) necessary to align the arm with that point is then determined, along with a common set of joint angles for Joints 1 – 5 that position the gripper to seize any of the objects. Angles for Joint 6 that

controls the closing and opening of the gripper to grasp and release an object were then identified. Finally a neutral position to which the arm could be returned in between movements was defined. The system was thus equipped with a set of primitives that could be combined to position the robot at any of the 6 grasping locations, grasp the corresponding object, move to a new position, and place the object there.

Cooperation Control Architecture: The spoken language control architecture illustrated in Fig 8.II is implemented with the CSLU Rapid Application Development toolkit (http://cslu.cse.ogi.edu/toolkit/). This system provides a state-based dialog management system that allows interaction with the robot (via the serial port controller) and with the vision processing system (via file i/o). It also provides the spoken language interface that allows the user to determine what mode of operation he and the robot will work in, and to manage the interaction via spoken words and sentences.

Figure 8.II illustrates the flow of control of the interaction management. In the Start state the system first visually observes where all of the objects are currently located. From the start state, the system allows the user to specify if he wants to ask the robot to perform actions (Act), to imitate the user, or to play (Imitate/Play). In the Act state, the user can specify actions of the form "Put the dog next to the rose" and a grammatical construction template is used to extract the action that the robot then performs. In the Imitate state, the robot first verifies the current state (Update World) and then invites the user to demonstrate an action (Invite Action). The user shows the robot one action. The robot re-observes the world and detects the action based on changes detected (Detect Action). This action is then saved and transmitted (via Play the Plan with Robot as Agent) to execution (Execute action). A predicate(argument) representation of the form Move(object, landmark) is used both for action observation and execution.

Imitation is thus a minimal case of Playing in which the "game" is a single action executed by the robot. In the more general case, the user can demonstrate multiple successive actions, and indicate the agent (by saying "You/I do this") for each action. The resulting intentional plan specifies what is to be done by whom. When the user specifies that the plan is finished, the system moves to the Save Plan, and then to the Play Plan states. For each action, the system recalls whether it is to be executed by the robot or the user. Robot execution takes the standard Execute Action pathway. User execution performs a check (based on user response) concerning whether the action was correctly performed or not. If the user action is not performed, then the robot communicates with the user, and performs the action itself. Thus, "helping" was implemented by combining an evaluation of the user action, with the existing capability to perform a stored action representation.

8. Experimental Results Part 2

For each of the 6 following experiments, equivalent variants were repeated at least ten times to demonstrate the generalized capability and robustness of the system. In less than 5 percent of the trials, errors of two types were observed to occur. Speech errors resulted from a failure in the voice recognition, and were recovered from by the command validation check (Robot: "Did you say …?"). Visual image recognition errors occurred when the objects were rotated beyond 20° from their upright position. These errors were identified when the user detected that an object that should be seen was not reported as visible by the system, and were corrected by the user re-placing the object and asking the system to "look again". At the beginning of each trial the system first queries the vision system, and updates the

World Model with the position of all visible objects. It then informs the user of the locations of the different objects, for example "The dog is next to the lock, the horse is next to the lion." It then asks the user "Do you want me to act, imitate, play or look again?", and the user responds with one of the action-related options, or with "look again if the scene is not described correctly.

Validation of Sensorimotor Control: In this experiment, the user says that he wants the "Act" state (Fig 8.II), and then uses spoken commands such as "Put the horse next to the hammer". Recall that the horse is among the moveable objects, and hammer is among the fixed landmarks. The robot requests confirmation and then extracts the predicate-argument representation - *Move(X to Y)* - of the sentence based on grammatical construction templates. In the Execute Action state, the action *Move(X to Y)* is decomposed into two components of *Get(X)*, and *Place-At(Y)*. *Get(X)* queries the World Model in order to localize X with respect to the different landmarks, and then performs a grasp at the corresponding landmark target location. Likewise, *Place-At(Y)* simply performs a transport to target location Y and releases the object. Decomposing the *get* and *place* functions allows the composition of all possible combinations in the *Move(X to Y)* space. Ten trials were performed moving the four object to and from different landmark locations. Experiment 1 thus demonstrates (1) the ability to transform a spoken sentence into a Move(X to Y) command, (2) the ability to perform visual localization of the target object, and (3) the sensory-motor ability to grasp the object and put it at the specified location. In ten experimental runs, the system performed correctly.

Imitation: In this experiment the user chooses the "imitate" state. As stated above, imitation is centered on the achieved ends – in terms of observed changes in state – rather than the means towards these ends. Before the user performs the demonstration of the action to be imitated, the robot queries the vision system, and updates the World Model (Update World in Fig 8.II) and then invites the user to demonstrate an action. The robot pauses, and then again queries the vision system and continues to query until it detects a difference between the currently perceived world state and the previously stored World Model (in State Comparator of Fig 1, and Detect Action in Fig 8.II), corresponding to an object displacement. Extracting the identity of the displaced object, and its new location (with respect to the nearest landmark) allows the formation of an *Move(object, location)* action representation. Before imitating, the robot operates on this representation with a meaning-to-sentence construction in order to verify the action to the user, as in "Did you put the dog next to the rose?" It then asks the user to put things back as they were so that it can perform the imitation. At this point, the action is executed (Execute Action in Fig 8.II). In ten experimental runs the system performed correctly. This demonstrates (1) the ability of the system to detect the goals of user-generated actions based on visually perceived state changes, and (2) the utility of a common representation of action for perception, description and execution.

A Cooperative Game: The cooperative game is similar to imitation, except that there is a sequence of actions (rather than just one), and the actions can be effected by either the user or the robot in a cooperative manner. In this experiment, the user responds to the system request and enters the "play" state. In what corresponds to the demonstration in Warneken et al. (2006) the robot invites the user to start showing how the game works. The user then begins to perform a sequence of actions. For each action, the user specifies who does the action, i.e. either "you do this" or "I do this". The intentional plan is thus stored as a sequence of action-agent pairs, where each action is the movement of an object to a particular target location. In Fig 6, the resulting interleaved sequence is stored as the "We

intention", i.e. an action sequence in which there are different agents for different actions. When the user is finished he says "play the game". The robot then begins to execute the stored intentional plan. During the execution, the "We intention" is decomposed into the components for the robot (Me Intention) and the human (You intention).

In one run, during the demonstration, the user said "I do this" and moved the horse from the lock location to the rose location. He then said "you do this" and moved the horse back to the lock location. After each move, the robot asks "Another move, or shall we play the game?". When the user is finished demonstrating the game, he replies "Play the game." During the playing of this game, the robot announced "Now user puts the horse by the rose". The user then performed this movement. The robot then asked the user "Is it OK?" to which the user replied "Yes". The robot then announced "Now robot puts the horse by the lock" and performed the action. In two experimental runs of different demonstrations, and 5 runs each of the two demonstrated games, the system performed correctly. This demonstrates that the system can learn a simple intentional plan as a stored action sequence in which the human and the robot are agents in the respective actions.

Interrupting a Cooperative Game: In this experiment, everything proceeds as in the previous experiment, except that after one correct repetition of the game, in the next repetition, when the robot announced "Now user puts the horse by the rose" the user did nothing. The robot asked "Is it OK" and during a 15 second delay, the user replied "no". The robot then said "Let me help you" and executed the move of the horse to the rose. Play then continued for the remaining move of the robot. This illustrates how the robot's stored representation of the action that was to be performed by the user allowed the robot to "help" the user.

A More Complex Game: In order to more explicitly test the intentional sequencing capability of the system, this experiment replicates the Cooperative Game experiment but with a more complex task, illustrated in Figure 7. In this game (Table 5), the user starts by moving0 the dog, and after each move the robot "chases" the dog with the horse, till they both return to their starting places.

Action	**User identifies agent**	**User Demonstrates Action**	**Ref in Figure 7**
1.	I do this	Move dog from the lock to the rose	B
2.	You do this	Move the horse from the lion to the lock	B
3.	I do this	Move the dog from the rose to the hammer	C
4.	You do this	Move the horse from the lock to the rose	C
5.	You do this	Move the horse from the rose to the lion	D
6.	I do this	Move the dog from the hammer to the lock	D

Table 5. Cooperative "horse chase the dog" game specified by the user in terms of who does the action (indicated by saying) and what the action is (indicated by demonstration). Illustrated in Figure 7.

As in the simplified cooperative game, the successive actions are visually recognized and stored in the shared "We Intention" representation. Once the user says "Play the game", the final sequence is stored, and then during the execution, the shared sequence is decomposed into the robot and user components based on the agent associated with each action. When the user is the agent, the system invites the user to make the next move, and verifies (by

asking) if the move was ok. When the system is the agent, the robot executes the movement. After each move the World Model is updated. As before two different complex games were learned, and each one "played" 5 times. This illustrates the learning by demonstration (Zollner et al. 2004) of a complex intentional plan in which the human and the robot are agents in a coordinated and cooperative activity.

Interrupting the Complex Game: As in Experiment 4, the objective was to verify that the robot would take over if the human had a problem. In the current experiment this capability is verified in a more complex setting. Thus, when the user is making the final movement of the dog back to the "lock" location, he fails to perform correctly, and indicates this to the robot. When the robot detects failure, it reengages the user with spoken language, and then offers to fill in for the user. This is illustrated in Figure 7H. This demonstrates the generalized ability to help that can occur whenever the robot detects the user is in trouble. These results were presented in Dominey (2007).

9. Discussion

This beginning of the 21st century marks a period where humanoid robot mechatronics and the study of human and artificial cognitive systems come in parallel to a level of maturity sufficient for significant progress to be made in making these robots more human-like in there interactions. In this context, two domains of interaction that humans exploit with great fidelity are spoken language, and the visual ability to observe and understand intentional action. A good deal of research effort has been dedicated to the specification and implementation of spoken language systems for human-robot interaction (Crangle & Suppes 1994, Lauria et al. 2002, Severinson-Eklund 2003, Kyriacou et al. 2005, Mavrides & Roy 2006). The research described in the current chapter extends these approaches with a Spoken Language Programming system that allows a more detailed specification of conditional execution, and by using language as a compliment to vision-based action perception as a mechanism for indicating how things are to be done, in the context of cooperative, turn-taking behavior.

The abilities to observe an action, determine its goal and attribute this to another agent are all clearly important aspects of the human ability to cooperate with others. Recent research in robot imitation (Oztop et al. 2006, Nehaniv & Dautenhahn 2007, Billard & Schaal 2006) and programming by demonstration (Zollner et al. 2004) begins to address these issues. Such research must directly address the question of how to determine what to imitate. Carpenter and Call (2007) The current research demonstrates how these capabilities can contribute to the "social" behavior of learning to play a cooperative game, playing the game, and helping another player who has gotten stuck in the game, as displayed in 18-24 month children (Werneken et al. 2006, Werneken & Tomasello 2006). While the primitive bases of such behavior is visible in chimps, its full expression is uniquely human. As such, it can be considered a crucial component of human-like behavior for robots (Carpenter & Call 2007).

The current research is part of an ongoing effort to understand aspects of human social cognition by bridging the gap between cognitive neuroscience, simulation and robotics (Dominey 2003, 2005, et al. 2004, 2006, 2007; Dominey & Boucher 2005), with a focus on the role of language. The experiments presented here indicate that functional requirements derived from human child behavior and neurophysiological constraints can be used to define a system that displays some interesting capabilities for cooperative behavior in the context of spoken language and imitation. Likewise, they indicate that evaluation of

another's progress, combined with a representation of his/her failed goal provides the basis for the human characteristic of "helping." This may be of interest to developmental scientists, and the potential collaboration between these two fields of cognitive robotics and human cognitive development is promising. The developmental cognition literature lays out a virtual roadmap for robot cognitive development (Dominey 2005, Werneken et al. 2006). In this context, we are currently investigating the development of hierarchical means-end action sequences. At each step, the objective will be to identify the behavior characteristic and to implement it in the most economic manner in this continuously developing system for human-robot cooperation.

At least two natural extensions to the current system can be considered. The first involves the possibility for changes in perspective. In the experiments of Warneken et al. the child watched two adults perform a coordinated task (one adult launching the block down the tube, and the other catching the block). At 24 months, the child can thus observe the two roles being played out, and then step into either role. This indicates a "bird's eye view" representation of the cooperation, in which rather than assigning "me" and "other" agent roles from the outset, the child represents the two distinct agents A and B for each action in the cooperative sequence. Then, once the perspective shift is established (by the adult taking one of the roles, or letting the child choose one) the roles A and B are assigned to me and you (or vice versa) as appropriate.

This actually represents a minimal change to our current system. First, rather than assigning the "you" "me" roles in the We Intention at the outset, these should be assigned as A and B. Then, once the decision is made as to the mapping of A and B onto robot and user, these agent values will then be assigned accordingly. Second, rather than having the user tell the robot "you do this" and "I do this" the vision system can be modified to recognize different agents who can be identified by saying their name as they act, or via visually identified cues on their acting hands.

The second issue has to do with inferring intentions. The current research addresses one cooperative activity at a time, but nothing prevents the system from storing multiple such intentional plans in a repertory (IntRep in Fig 6). In this case, as the user begins to perform a sequence of actions involving himself and the robot, the robot can compare this ongoing sequence to the initial subsequences of all stored sequences in the IntRep. In case of a match, the robot can retrieve the matching sequence, and infer that it is this that the user wants to perform. This can be confirmed with the user and thus provides the basis for a potentially useful form of learning for cooperative activity.

In conclusion, the current research has attempted to build and test a robotic system for interaction with humans, based on behavioral and neurophysiological requirements derived from the respective literatures. The interaction involves spoken language and the performance and observation of actions in the context of cooperative action. The experimental results demonstrate a rich set of capabilities for robot perception and subsequent use of cooperative action plans in the context of human-robot cooperation. This work thus extends the imitation paradigm into that of sequential behavior, in which the learned intentional action sequences are made up of interlaced action sequences performed in cooperative alternation by the human and robot. While many technical aspects of robotics (including visuomotor coordination and vision) have been simplified, it is hoped that the contribution to the study of imitation and cooperative activity is of some value.

Acknowledgements: I thank Jean-Paul Laumond, Eiichi Yoshida and Anthony Mallet from the LAAS Toulouse for cooperation with the HRP-2 as part of the French-Japanese Joint

Robotics Laboratory (AIST-Japan, CNRS-France). I thank Mike Tomasello, Felix Warneken, Malinda Carpenter and Elena Lieven for useful discussions during a visit to the MPI EVA in Leipzig concerning shared intentions; and Giacomo Rizzolatti for insightful discussion concerning the neurophysiology of sequence imitation at the IEEE Humanoids meeting in Genoa 2006. This research is supported in part by the French Minister of Research under grant ACI-TTT, and by the LAFMI.

10. References

Bekkering H, WohlschlagerA , Gattis M (2000) Imitation of Gestures in Children is Goal-directed, The Quarterly Journal of Experimental Psychology: Section A, 53, 153-164

Billard A, Schaal (2006) Special Issue: The Brain Mechanisms of Imitation Learning, Neural Networks, 19(1) 251-338

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

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.

Carpenter M, Call Josep (2007) The question of 'what to imitate': inferring goals and intentions from demonstrations, in Chrystopher L. Nehaniv and Kerstin Dautenhahn Eds, *Imitation and Social Learning in Robots, Human sand Animals*, Cambridge Univerity Press, Cambridge.

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

Cuijpers RH, van Schie HT, Koppen M, Erlhagen W, Bekkering H (2006) Goals and means in action observation: A computational approach, *Neural Networks* 19, 311-322,

di Pellegrino G, Fadiga L, Fogassi L, Gallese V, Rizzolatti G (1992) Understanding motor events: a neurophysiological study. *Exp Brain Res.*;91(1):176-80.

Dominey PF (2005) Toward a construction-based account of shared intentions in social cognition. Comment on Tomasello et al. 2005, *Beh Brain Sci*. 28:5, p. 696.

Dominey PF, (2003) Learning grammatical constructions from narrated video events for human–robot interaction. *Proceedings IEEE Humanoid Robotics Conference*, Karlsruhe, Germany

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.

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

Dominey PF, 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*.

Dominey PF, Hoen M, Inui T. (2006) A neurolinguistic model of grammatical construction processing. *Journal of Cognitive Neuroscience*.18(12):2088-107.

Dominey PF, Mallet A, Yoshida E (2007) Progress in Spoken Language Programming of the HRP-2 Humanoid, *Proc. ICRA 2007, Rome*

Dominey PF (2007) Sharing Intentional Plans for Imitation and Cooperation: Integrating Clues from Child Developments and Neurophysiology into Robotics, Proceedings of the AISB 2007 Workshop on Imitation.

Fong T, Nourbakhsh I, Dautenhaln K (2003) A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42 3-4, 143-166.

Goga, I., Billard, A. (2005), Development of goal-directed imitation, object manipulation and language in humans and robots. In M. A. Arbib (ed.), *Action to Language via the Mirror Neuron System*, Cambridge University Press (in press).

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

Kozima H., Yano H. (2001) A robot that learns to communicate with human caregivers, in: *Proceedings of the International Workshop on Epigenetic Robotics*,.

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

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

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)

Nehaniv CL, Dautenhahn K eds. (2007) *Imitation and Social Learing in Robots, Humans and Animals*, Cambridge University Press, Cambridge.

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

Oztop E, Kawato M, Arbib M (2006) Mirror neurons and imitation: A computationally guided review. *Neural Networks*, (19) 254-271

Pickering MJ, Garrod S. (2004) Toward a mechanistic psychology of dialogue.Behav Brain Sci. Apr;27(2):169-90

Rizzolatti G, Craighero L (2004) The Mirror-Neuron system, *Annu. Rev. Neuroscience* (27) 169-192

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

Sommerville A, Woodward AL (2005) Pulling out the intentional structure of action: the relation between action processing and action production in infancy. *Cognition*, 95, 1-30.

Tomasello M, Carpenter M, Cal J, Behne T, Moll HY (2005) Understanding and sharing intentions: The origins of cultural cognition, *Beh. Brain Sc;*. 28; 675-735.

Torrey C, Powers A, Marge M, Fussel SR, Kiesker S (2005) Effects of Adaptive Robot Dialogue on Information Exchange and Social Relations, Proceedings HRI 2005.

Warneken F, Chen F, Tomasello M (2006) Cooperative Activities in Young Children and Chimpanzees, *Child Development*, 77(3) 640-663.

Warneken F, Tomasello M (2006) Altruistic helping in human infants and young chimpanzees, *Science*, 311, 1301-1303

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).

Humanoid Robots: New Developments

Edited by Armando Carlos de Pina Filho

ISBN 978-3-902613-00-4

Hard cover, 582 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

For many years, the human being has been trying, in all ways, to recreate the complex mechanisms that form the human body. Such task is extremely complicated and the results are not totally satisfactory. However, with increasing technological advances based on theoretical and experimental researches, man gets, in a way, to copy or to imitate some systems of the human body. These researches not only intended to create humanoid robots, great part of them constituting autonomous systems, but also, in some way, to offer a higher knowledge of the systems that form the human body, objectifying possible applications in the technology of rehabilitation of human beings, gathering in a whole studies related not only to Robotics, but also to Biomechanics, Biomimmetics, Cybernetics, among other areas. This book presents a series of researches inspired by this ideal, carried through by various researchers worldwide, looking for to analyze and to discuss diverse subjects related to humanoid robots. The presented contributions explore aspects about robotic hands, learning, language, vision and locomotion.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen