Human-Robot Cooperation Based on Interaction Learning

Lallee S.[1], Yoshida E.[2], Mallet A.[2], Nori F.[3], Natale L.[3], Metta G.[3], Warneken F.[4], Dominey P.F.[1]

1 - Stem Cell and Brain Research Institute, INSERM U846. 18 avenue Doyen Lépine ,69675 Bron Cedex, France.

2 – Laboratoire d'Analyse et Architecture des Systèmes, CNRS. 7 avenue du Colonel Roche, 31077 Toulouse, France.

3 - Italian Institute of Technology, Central Research Labs, Genoa Headquarter. Via Morego, 30. Genoa, Italy.

4 - Department of Developmental and Comparative Psychology, Max Planck Institute for Evolutionary Anthropology. Deutscher Platz 6, D-04103 Leipzig, Germany.

1 Introduction

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 stated in [1] : "the user should not have to become a programmer, or rely on a programmer, to alter the robot's behavior, and the user should not have to learn specialized technical vocabulary to request action from a robot." To achieve this goal, one option is to consider the robot as a human apprentice and to have it learn through its interaction with a human. This chapter reviews our approach to this problem.

An apprentice is an able-bodied individual that should interactively assist an expert, and through this interaction, they should acquire knowledge and skill in the given task domain. In other words, the expert teaches the apprentice by sharing a task with him, or by direct demonstration or explanation of what to do. In this context, the robot owns a repertoire of useful actions which execution can be requested by the user. While the human uses theses actions in order to achieve task, the robot should extract information about the human goal and how actions can imbricate to reach this goal.

The implementation of this apprentice architecture is one of our long term goal for which we developed the Spoken Language Programming system (SLP). First ([2],[3]) we established a primary mapping between sentences and actions, allowing verbal command of a robot and online creation of new commands. By including some visually guided actions this provides already a large repertoire of behaviors. However, even if the SLP allows fluent commanding of the robot, everything that the robot does has first to be predefined by the user. This is the reason that leads us to add anticipation ability to the system in [4]. This ability allows the robot to not only wait for commands, but to predict or even execute these commands without any order received from the human. In this chapter, we will review our previous work relative to the SLP development and extend it so that the user and the robot will have shared representations not only for actions, but also for objects and high level tasks that can imply sub-tasks. These shared representations and “hierarchical actions” are part of a more global cooperation skill used by humans which is described in [5]. In the second part of this chapter we will describe some results about the human cooperation ability and how we can use them to improve the SLP so it will provide to robots a human like cooperation ability.

1.1 Linking words to actions – Spoken Language Programming

Robots will integrate gradually everyday life over the next century. It will require a way for non specialists to use them for a wide range of tasks that are not predefined and that will occur in an unknown environment. The most common ability used by humans to communicate is spoken language; so it can provide a very rich vector for communication between a user and a robot. Through grammatical constructions, complex meanings can be communicated. Construction grammar (CxG) provides a linguistic formalism for achieving the required link from language to meaning [6].Meaning is represented in a predicate-argument structure as in [6], based on generalized abstract structures as in [7]. 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 Predicate - Argument structures to extract robot commands from natural language, and to generate natural language descriptions of physical events extracted from video scenes ([7-12]). In this section we review our work [2, 3, 7] aimed to the devellopment of a Spoken Language Programming system (SLP). The objective of the SLP is to to use natural language in order to allow human users to both command, program or teach the robot with spoken language. In a related context, Nicolescu and Mataric [12] 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. [13] asked naïves 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 [13]. This indicates the importance of implementing the mapping between language and behavioral primitives for natural language instruction or programming [14, 15]. Learning by imitation and/or demonstration likewise provide methods for humans to transmit desired behavior to robots [11]. The SLP extends such methods in a complimentary way. We established a representative scenario where human and robot have to interact using spoken language in order to accomplish a task. We will first present this scenario and the robotic platform requirements for it execution, and then we will use it to benchmark successive versions of the SLP.

1.2 A scenario for human-robot cooperation

In order to test the SLP a task that involve human – robot cooperation has to be defined. We first introduced the Cooperative Table Construction Task in [2] and refined it in [3, 4]. In the Cooperative Table Construction Task a robot and a human have to cooperate in order to build a small table (Fig. 1). The user has first to ask the robot for the table's legs, and then he will have to screw then to the table's surface which will be hold by the robot. All the interaction between the two participants are done using spoken language. One major interest of this task is that its execution includes regularities : the same sequence of actions will occur several times. For example, the user will have first to ask "grasp a leg", then "pass it to me" and finally "hold the surface". Since this sequence will be the same for the 4 legs, it can be easily extracted by the user, which will be able to speed up the task by programming the robot. Moreover, this task involves a lot of knowledge that can be hard coded or learned by the robot in real time. It involves mainly two domains: mapping words to actions (what does "pass" mean?) and mapping words to arguments (what is a "leg"?), the combination of these two mappings allows the execution of a concrete physical action based on a Predicate - Argument structure extracted from spoken language.

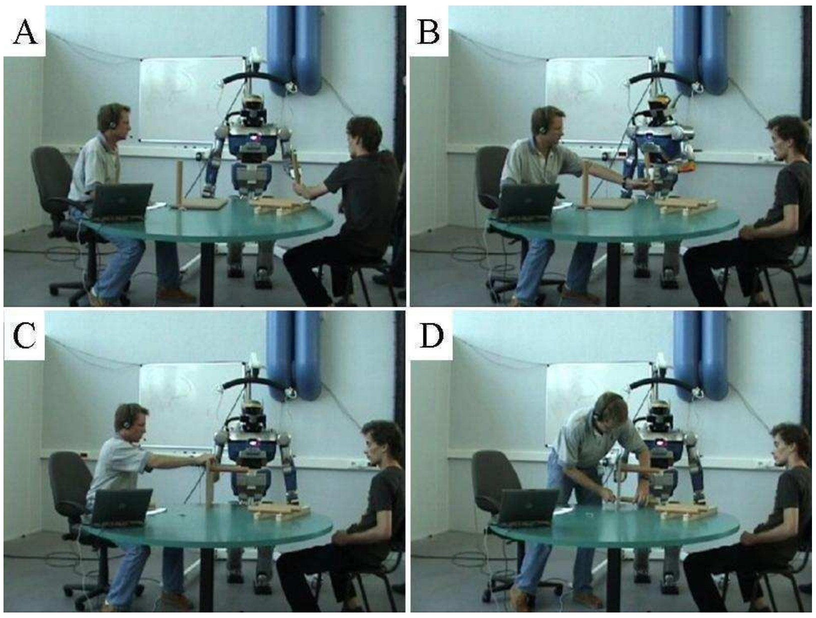

Fig. 1 . The Cooperative Table Construction Task with HRP-2 : robot and human have to cooperate in order to build a table (A,D). The robot has to pass legs to the user (B) and hold the table while the user screws the legs (C).

1.3 The robotic platforms

One of the main interests of the SLP is that it is not robot specific. The speech interface (synthesis & recognition) is provided by the CSLU Rapid Application Development (RAD) Toolkit (http://cslu.cse.ogi.edu/toolkit/) which runs on an independent computer remotely connected to the robot. 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. It also allows scripting in TCL which allows implementing complex behaviors. The vision system is also robot independent and need only that a YARP interface (http://eris.liralab.it/yarp/) has been defined to access the robots cameras. Then the object recognition is then provided using Spikenet, a commercial system based on research of Thorpe and al. [16] related to vision processing in the cotex. The only components of the SLP that are robot specifics are the 3D object localization (since all robots don't have the same camera system) and the basic set of atomic actions. Since all robots have different bodies and abilities we assume that they provide some basic actions that the SLP will be able to call (e.g: "grasp" will not be the same according to whether you are using a humanoid robot or a dog robot.). Anyway, these requirement can be easily implemented on any robot and we already used successfully the SLP on multiple robot platforms including :

- Kawada Industries HRP-2 humanoid robot [17] under the control of the OpenHRP controller [18]
- AIBO ERS7 with a WIFI interface
- Lynxmotion 6DOF robot arm
- Khepera mobile robots with a serial port controller [8]
- iCub humanoid robot [19] using a YARP [20] based architecture

However, the Cooperative Table Construction Task scenario is more suited to a humanoid robot, so most of the researches presented in this chapter are made using the HRP-2 or the iCub which are robots with many effective degrees of freedom (respectively 30 and 54) and possibilities for rich cooperative interaction.

2 The Development of Spoken Language Programming

The SLP is not static and has significantly evolved since its creation. The first implementation was a simple system that allowed the creation of macros, this system was then refined to turn these macros into procedures and to include the use of vision and arguments. The most recent evolution, presented here for the first time, allows the creation of hierarchical functions (functions that call functions) and the establishment of a common vocabulary between the robot and the user regarding visual object names. In this section we will present all these successive evolutions of the SLP and suggest what can be improved in future work.

2.1 Macro style programming

The first implementation of SLP is described in [2]. In this study we used the HRP-2 and an early version of the Cooperative Table Construction Task. There was no vision included in the system and only a small set of available actions. However this provided already enough material to achieve a proof of concept for the SLP.

2.1.1 Atomic actions and learning commands

The central idea in SLP is compositionality: based on a finite set of atomic action primitives the user should be able to compose arbitrary new cooperation behaviors. As we stated above, one of the few robot specific components of the system is the set of atomic actions. These are actions which are available to the user without any need of programming, in the case of the HRP2 theses actions were a set of static postures corresponding to required function in the task (table 1). The set of atomic actions for this study is presented in table 1. Each atomic action is called by the user through the SLP using a verbal command, thus allowing the user to control the robot using speech.

Verbal 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)
OpenRight	Open right hand (1 DOF)
Give it to me	Rotate hip to pass the object in left hand to User (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 1: HRP-2 Set of Atomic Actions

In addition to the set of atomic actions, 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.

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.

Table 2: SLP Learning and control commands

By using both the set of atomic actions and the learning commands, the user is able to make the task execution easier and faster as we show in the next section.

2.1.2 Experimental results

In order to evaluate the implemented system, we used the Cooperative Table Construction Task in two different ways. First the user and the robot had to assemble the table, and then they had to cooperate again but 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 atomic actions was 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.

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.**

The user begins its first interaction with the robot. He 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.

<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. <A third party user places the table leg in the robot's left hand>
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.
Left open.
You said left open? Yes.

Opening left hand. <As the Robot opens the hand, User 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 User 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, User places the table top within the robot's grasp>
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, User attaches the leg to the table and then gives the "continue" command when ready.>
Continue.
I am ready.
Right open.
You said right open? Yes.
Opening right hand. <User takes the table from the robot and prepares for the next leg.>
I am ready.

At this point, the user 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 he has gained, he is now ready to "program" the robot for the repetitive component of the task by using the "learn" command.

Learn.
You said learn? Yes. *I'll start learning now. I am ready.*
Prepare. <The interaction continues as for the previous leg until the end >
Continue.
I am ready.
Right open.
You said right open? **Yes**.

Opening right hand. <User takes the table>
I am ready.
OK.
You said OK? **Yes**. <"OK" indicates the macro end (table 2) >
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.

Macro
You said macro ? Yes.
Running the macro.
Preparing to grasp.
Waiting for your signal.
Continue
Closing left hand. <Robot takes leg>
Giving you the object.
Opening left hand. <User takes leg>
Getting ready to hold.
Waiting for your signal.
Continue
Closing right hand.
Waiting for your signal <User attaches the leg>
Continue
Opening right hand. In line macro finished. I am ready. < >

Third leg is now attached. User takes the table. Same procedure with macro for attaching fourth leg.

Performance Analysis of SLP Effects

During the assembly of the first two legs, the user 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 2 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.

Disassembling the table

Here we test 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. User will first ask the robot to hold the table while he removes the leg. He 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 the third party user 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.

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.

As in the previous experiment, after one run with the first leg, the user is now ready to program the robot. Again, he initiates the program storage by saying "Learn" and then executes step-by-step the procedure for taking a leg off and passing it to the third party user 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.

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. >

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). Fig 2 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 ($F(1,6) = 277$, $p < 0.0001$).

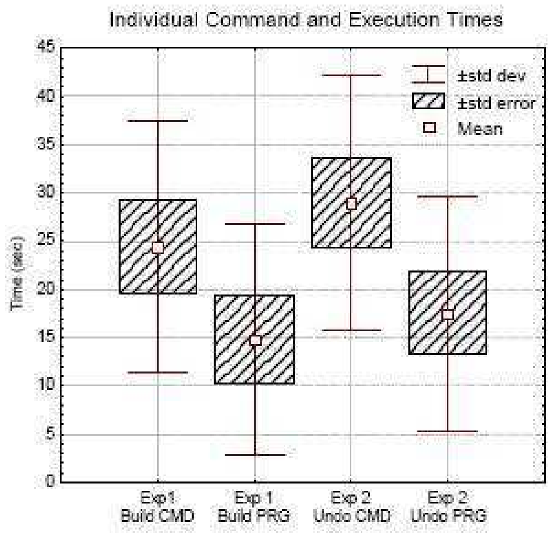

Fig.2. Average command execution times for the Building (Exp1) and Undoing (Exp2) task using spoken language for on-line commanding (CMD) and for macro programming (PRG)

2.1.3 Conclusion

Despite the speed improvement induced by the SLP 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 [21] 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. 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 [15, 22]. Language has been used in this context for correcting and clarifying what is being learned by demonstration [12]. 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 [13, 23]. 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 [14]. 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 [8, 9]. In this context of language-based human-robot cooperation, this research demonstrated - 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.

2.2 Vision based procedures with arguments

The first implementation of the SLP was limited: multiple macros were not allowed, and the actions offered were quite rigid. The first refinement of the SLP has been done in [3]: we integrated vision and motion planning into the SLP framework, providing a new level of flexibility in the behavior that can be created. Most important we possibility to the user to create “generic” functions with arguments (e.g. Give me X), and we allowed multiple function creations. We thus demonstrated again with the HRP-2 equipped with vision based grasping the ability to acquire multiple sensory motor behavioral procedures in real-time through SLP in the context of a cooperative task. The humanoid robot thus acquired new sensory motor skills that significantly facilitate the cooperative human-robot interaction.

2.2.1 Adding visually guided action to the robot

One of the most impressive improvements of the SLP in [3] is the fact that the robot is able to localize an object based on vision and to reach/grasp it wherever it is in the reachable workspace. In the original study, we used the OpenCV library and some basic color recognition algorithm in order to recognize the different colored legs. We then used stereovision triangulation to get the object coordinates in a 3D space. However, the only requirement of the vision module is to provide 3D coordinates of the object to the SLP, it can be achieved in different ways. For example, in our latest studies we use a commercial program (Spikenet) [16, 24] which allows the recognition of arbitrary objects based on their shape; but one can think about using other 3D localization systems using markers, etc.

The second requirement is to be able to use these 3D coordinates in order to guide certain actions. Basic actions that involve vision are for example turning the head

to gaze at an object, or moving the arm and hand in order to grasp it. To do so we used an inverse kinematic solver which gives, for a given 3D position, the values of the arm's joints angles which will bring the hand to this position. This approach is simple but sufficient for our purposes, however, usage of more complex motor control algorithm is allowed but they are behind the scope of our research.
Combining object vision and inverse kinematics thus provides the robot with a behavioral capability to localize a specified object and grasp it. This allowed the creation of an extended set of atomic action described in table 3.

Motor Command	Resulting Actions
Ready Position	Move bith arms to a ready position above the workspace (6DOF)
Take the $VAR1 leg.	Visually localize the $VAR1 colored object, and grasp it. $VAR1 = {green, yellow, rose, orange}. (6DOF)
Open(Right,Left)	Open right/left hand (1DOF)
Close(Right,Left)	Close right/left hand (1DOF)
Turn(Right,Left,Center)	Rotate chest right, left or center (1DOF)
Reach(Right,Left)	Extend right/left arm (5DOF)

Table .3. Set of atomic action. “Take the leg” is a vision-based action.

2.2.2 Learning arguments taking procedures

In the previous study, the learned programs were fixed sequences of actions, and thus had strong requirements on the invariant conditions of execution. In [3] we extended such methods in a complimentary way by using spoken language to create complex motor procedures or behaviors that are flexible to changes in the environment. This robustness is provided through the learning of procedures (i.e. actions) that take arguments (i.e. objects that are manipulated in the action). The learned behaviors correspond to procedures that take arguments, e.g. “Give me X”, where the robot uses vision and motion planning to localize X which can be arbitrarily located within the robots perceivable works space. As in the previous study, this procedures definition is controlled by a set of learning and interaction commands presented in table 4. The updated version of the SLP allowed multiple procedures to be created, and allows these procedures to take arguments. While the general solution is to allow an arbitrary number of procedures to be created and named, we will describe only two specific procedures in this section: “**Give me the X**” and “**Hold this**”.

When the user issues a command such as “**give me the green leg**”, the SLP determines if it has a corresponding procedure, by looking up that command in the command database. If there is no procedure defined, the robot asks the user to "*show me how to give you the green block*". The user proceeds to issue action commands to achieve the desired task that will make up this sequence. Of particular interest in this case, if the user issues a command with an argument as a component of the new behavior to be learned, such as "grasp the green leg," then the argument to the grasp command becomes bound to the argument for the give command. 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 "**give me the rose leg**" is issued, this file is read into an array, and the commands are sequentially executed. Importantly, the value "rose" is bound to variable $VAR1 which is passed to any commands that take arguments in the procedure, notably the "**take the X leg**" command. This "natural" programming of argument-taking procedures gives the user a powerful tool for specifying generalized behaviors.

Commands	**Correspondence**
Give me the $VAR1 leg	Learned procedure
Hold this	Learned procedure
OK	Store encoded command sequence
Wait	Interrupt command execution until a spoken "continue" command is issued
Continue	Terminate the "wait" pause and resume execution

Table .4. Programming commands.

2.2.3 Experimental results

Here again we used the Table Construction Task as a benchmark for the SLP. We will describe first how the user teach the two procedures "give me X" and "hold this", then we will show how to use them and study the effect of these procedures on the task execution.

"*Give me the X" Procedure*

<Robot initiates the interaction> *Describe an action.*
Give me the green leg.
Please show me how to give you the green leg.
Take the green leg.
Attempting to grasp the green leg. <System executes the Vision and Motion Planning to place the hand above the leg.>
Is that good ? <Robot asks user if the hand is correctly placed>. Yes.
Attempting to grasp. <Robot descends hand, grasps the object and raises the hand – executing the Take command>

Describe an action.
Turn right. <Robot turns right>
Open right hand. <Robot opens right hand, giving object to user>.
Ready position. <Robot pulls right hand back to ready position>
OK - Finish learning.
You said OK, so I will store the learned program.

At this state in the dialog, the system writes the sequence of commands to a file, in the format: *Give me the $VAR1 leg { Take the $VAR1 leg Turn right Open right hand Ready position }* When the "give" procedure is subsequently invoked with an argument, as in "**give me the orange leg**", the word "orange" that is recognized in the $VAR1 position for the "give" command will be passed as an argument to the take command.

"Hold This" Procedure

At this point in the cooperative interaction, the user needs the robot to hold the table while he attaches the leg.

Hold this <Robot does not have a stored procedure for "hold this">
Can you show me how to hold?
Reach left <Robot extends left arm, and user places table in the hand (Fig 2C)>.
Close left hand <Robot closes left hand to hold the table (Fig 2C)>
Wait <User requests robot to wait while he attaches the leg to the table.>
Continue <User indicates he is finished>
Open left hand <Robot opens left hand to release the table (Fig 2C)>
OK - Finish learning
You said OK, so I will store the learned program.

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

Execution of Learned Procedures

Now the user can apply these learned procedures for the 2-4th legs.

Give me the orange leg <System executes the learned procedure to place the hand above the orange leg.>
Is that good? <Robot asks user if the hand is correctly placed>
Yes.

Attempting to grasp <Robot descends hand, grasps the object and raises the hand – executing the Take command>
Turning right
Opening right hand <to give the leg to the user>
Moving to ready position <"Give" procedure finished>
Describe an action
Hold this
Reaching left <Robot begins to execute the stored procedure and reaches with left hand>
Closing left hand <User places the table in the robot's hand, and robot closes hand to hold the table.>
Waiting for your signal <User attaches the leg and then indicates to the robot to go on.> **Continue**
Opening left hand <Robot releases the table, assumes the ready position and turns to the center>
Moving to ready position
Turning to center <Leg is now attached, using the "give" and "hold" procedures>

Performance Analysis of SLP Effects

In order to quantify the benefits of the learning capability, we can analyzed the number of commands and the time required to perform the "give" and "hold" procedures when each command was individually enumerated during learning, vs. when the learned procedures were employed. During the attachment of the first leg, the user issued 13 distinct commands, and getting the leg and holding the table took approximately 2:00 and 2:16 respectively. Once the procedures were learned, each of the remaining 3 legs was attached using only 4 commands, and the total execution time for the three legs was 5:24. In other words, this means that in a total task completion time of less than 10 minutes, the human was able to program two different behavior procedures and then immediately use them in order to assemble the table, yielding a significant reduction in the number of commands and execution time required.

Another important point that improves the performances on a long term point of view is that the user can create and store multiple procedures. These procedures will be available for further tasks: for example we can imagine that the "**Give me X**" procedure can be used for a drawing interaction (i.e: "**give me orange pen**") which will speed up even more the task because the robot will not have to learn again how to give an object.

2.2.4 Conclusion

In [3] we demonstrated how the functional vocabulary of the robot is extended by learning a new command and a new grammatical construction. Part of the richness of grammatical constructions in language is that they encode predicate – argument relations in a natural and understandable manner. These predicates can be complex relations between (or operations on) the set of accompanying arguments. This provides a vastly richer communication ability than does the use of single word commands. This allows the procedure to operate on an entire class of objects (i.e. those that can be "taken" by vision-based grasping capability), and makes the procedure robust to variability in the position of the argument objects. This implies more than just plugging together the specific components for vision, motor control, language and sequence learning. In particular, the "form to meaning mapping" aspect of construction grammar provides a principled method for linking of language to "meaning" in the context of predicate-argument meanings.

2.3 Flexible Common Ground

Even if the SLP as it is described in [2, 3] already provide nice programming possibilities to a naïve user, it is still possible to improve it in order to allow more powerful human robot interaction. The goal of the system is that the user can arrive with a new set of objects, and some ideas about the names for these objects, and what he wants the robot to do to help him. The construction of the common interaction framework can be divided in three different parts: setting visual recognition, teaching non predefined postures and finally having new complex behaviors learned by the robot. In this section we describe how any naive user can achieved each of these teaching processes.

2.3.1 Vocal Interaction Prerequisite

One of the main limitation of the SLP as it was until now, is that the user has to know in advance the robot's vocabulary. He had to know how were named the manipulated objects and the robot specifics actions. In this version, we propose to the user to be able to name these objects and also to create his own new postures. In order to do that, we have to address a technical limitation. To provide speech recognition, RAD needs to be fed by every word it can have to recognize. We do that using three text files (objects' names, postures' names, and behaviors' names) that contain all the words that one can use. These three files will describe the complete vocabulary of the SLP and can contain a huge set of words. However, they can be edited manually by the user if he needs to do so.

2.3.2 Real-Time Object Learning

As we mentioned in the previous section, we use for the vision of the SLP a system call Spikenet [16, 24]. The main advantage of this system is that it allows very fast object recognition based on template learning. In other words, you give to the system some models of the objects you want to use, and it recognize them in real time on the video stream from the webcam. First step is the creation of objects' models: the user displays the set of objects on the workspace and uses the mouse to circle on the camera windows the objects he wants the robot to recognize. During the spoken language interaction, the user then presents the objects one at a time, and tells the system the name of the object via spoken language as described in this sample.

Learn object names
Please, show me only one object <User removes all object from the workspace but one>
Ok.
I can see an object, please tell me its name.
Leg
So this is a leg. Do you want to teach me another object ?
No.
Learning finished, what should I do now ?
Describe the scene. <User ask the robot to describe what he see>
I can see a leg.

2.3.3 Improved Set of Atomic Actions

In previous studies, the set of atomic actions was completely static and it was not possible for the user to create his own postures. In this study we refined the SLP architecture to allow such a possibility. The set of atomic actions does not contain anymore fixed postures, but only atomic procedures that are defined for every robot and that can be used to build more complex procedures. We describe the set we used in our experiment on a humanoid torso in table 5. Note that for a complete humanoid robot this set can be much more extensive.

Atomic Action	**Correspondence**
Posture $VAR	Move the robot into the posture described by $VAR (see 2.3.3.4 Posture Creation)
Move <up/down> $VAR	Move the actual arm along the vertical axis by $VAR centimeters
Move <left/right> $VAR	Move the actual arm along the horizontal axis by $VAR centimeters
Move <in/out> $VAR	Move the actual arm along the depth axis by $VAR centimeters
Open/Close	Open or close the hand of the actual arm
Reach $VAR	Visually locate the object $VAR and reach it with the actual arm

Describe	The robot tell the name of every objects that are present on the workspace
Wait	Stop execution of all commands until a “Continue” is issued
Continue	Resume execution of commands when being in a “Wait” state.

Table.5. Actual version of the set of atomic action.

One can already notice that this set of atomic actions does not include the action “grasp X”. This is because “grasp X” is not atomic and is composed by: *{reach X, move down 5, close, move up 10, posture neutral position}*. We will see how to define such a composed action later. First we describe how user can create their own postures, like *neutral position*.

2.3.4 Posture creation

User can ask the robot to go to any posture by telling him “Posture Posture_Name”. In the case of an unknown posture, the robot goes in a “relative movement command” state. This state allow the user to teach the desired posture by verbally moving the gripper with the Move (up|down ; left|right ; in|out) commands (Table 5). This provides a highly flexible method to create useful postures (e.g. for going to a neutral position)

Posture neutral position
Sorry, I don't know what is neutral position . Do you want to teach it to me?
Yes
Where do you want me to go?
Move up 20
Did you say to move up of 20 centimeters?
Yes <Robot moves>
Where do you want me to go?
Move right 20
Did you say to move right of 20 degree?
Yes <Robot moves>
Where do you want me to go?
Save

The user thus positions the robot using spoken language in less than 30 seconds. The posture is then added to the robot knowledge and can be embedded in a behavioral sequence. Note that in the case of a posture definition, the absolute values for the joints angles are recorded as a posture, so asking the robot to reach this posture will put him exactly as it was (not just 20cm upper/right of its actual position).

2.3.5 Complex behavior creation

Even if the user is able to call the atomic actions from the set of atomic actions, it is much more valuable for him to define his own actions. We exploit simple grammatical constructions that allow the mapping of natural language sentences onto specification of procedures and their arguments. This improves the SLP capabilities and make it able provide more speed improvement. In this section we show how to create a simple behavior, and then we use more complex ones in order to benchmark the system.

Learning to Grasp X

As we mentioned before, the grasp X behavior can be easily defined by the user in the case of a robot working on a flat table. Of course the grasping function can be hardcoded in the robot, and then it should be part of the set of atomic actions, but in our case it's a useful way to illustrate the power of the new SLP architecture. Here is how we defined "Grasp X" for our experiments:

Grasp leg
Sorry, I don't know how to grasp. Do you want to teach it to me?
Yes
What should I do?
Reach leg
Reaching <Robot moves his hand over the table's leg>
What should I do?
Move down 5
<Robot lay his hand on the table's leg>
What should I do?
Close
<Robot closes his hand and grasp the object>
What should I do?
Move up 10
<Robot grasp the table's leg>
What should I do?
Posture neutral position
<Robot moves to the previously learned neutral position>
What should I do?
Save

The behavior "grasp X" is now saved in the robot knowledge. The SLP inferred that the procedure took an argument and it automatically found that this argument was the same the one for the atomic action "reach". So when the user will call "grasp ball", the system will look and reach the ball and not the leg. In a very few steps the user defined a behavior that will be useful in a lot of task, and he will even be able reuse it in more complex behaviors like “Put X at Y”.

Learning to Put X at Y

Here the user asks the robot to put the object “front” (it is the front part of a helicopter wood model) on a visual target labeled “3”. The system does not know “put”, and infers there are two arguments. As the user teaches the robot, known procedures “grasp” and “reach” have arguments that match the first and second arguments for “put” and so this argument binding order is established for the learned sequence. Note that the learning procedure takes only 71 seconds.

```
12:11:05 User : put front 3
12:11:07 Robot : Do you mean put front 3 ?
12:11:09 User : yes
12:11:16 Robot : Tell me what to do.
12:11:18 User : grasp front
12:11:22 Robot : Loading grasp front. Tell me what to do.
12:11:36 User : reach 3
12:11:39 Robot : Loading reach. Tell me what to do.
12:11:57 User : open
12:12:00 Robot : Loading open.  Tell me what to do.
12:12:07 User : move neutral
12:12:10 Robot : Loading move neutral. Tell me what to do.
12:12:16 User : save
```

Learning to Swap X and Y

Once the “put” procedure is learned it can be used in a more complex. “swap” procedure, which itself is learned in just over two minutes.

```
12:38:58 User : swap train front
12:39:11 Robot : Tell me what to do.
12:39:15 User : put train 3
12:39:19 Robot : Loading put train 3. Tell me what to do.
        …. Speech recognition error and recovery …
12:39:59 User : put front 1
12:40:02 Robot : Loading put front 1  Tell me what to do.
12:40:30 User : put train 2
12:40:34 Robot : Loading put train 2  Tell me what to do.
12:41:01 User : save
```

At this point the “swap” operation has been learned, using graphical markers (learned as any other object) as place holders for the objects to be swapped. The new behavior is then demonstrated to transfer to new objects, and the complex behavior is achieved in just slightly over 1 minute

12:42:10 User : **swap tail nose**
12:42:13 Robot : *Loading swap tail nose . Is it ok ?*
12:42:16 User : **yes**
12:43:15 Robot : *Tell me what to do.*

Then it's possible to use swap in an even more high level new sequence and so on. After a while of teaching the robot what he wants to do, the user is able to command the robot in a very efficient way without taking care of the most basic interactions.

2.4 Discussion

With the SLP implementation, we started to build a framework which facilitates the cooperation between human beings and robots. It allows any user to program automated behavior on any kind of robot in a very easy and natural manner. The robot is able to “learn” what the human wants to teach to him and then the user is able to take benefit of this knowledge in order to improve the way he uses the robot. Since the SLP is a kind of programming language, we can imagine multiple ways to improve it (e.g: adding “if…then” conditional commands, loops, etc.). However we can’t really call that a cooperative interaction: the robot interacts with the human but it does not share any goal with him, he doesn’t show any intentionality. Indeed, the SLP provides the user with a vocal puppet mastering ability but it doesn’t give to the robot any choice regarding his own behavior. A great improvement of the system could be to enable it to learn even without direct teaching, only by interacting with the user. Such an ability can really make robots useful to humans and avoid the “rigid” aspect of their behavior. In the second part of this chapter we will present our efforts to bring to the SLP this human-like cooperation ability.

3 Beyond SLP – Anticipation & Cooperation

In his well known test [25], Turing defined a way to test artificial intelligence by having a human being chatting with it. Spoken language is a hard problem; however we can generalize this test to any kind of collaborative ability between an artificial system and a human. When the artificial system owns a physical body, like a robot, the interactions framework provides a perfect platform to test how the system should behave. Collaborative tasks, like the Table Construction Task that we defined previously, are situations from which we can extract primates' specific behaviors. Such behaviors should be implemented into collaborative artificial systems like robots (especially humanoids one) if we want them to be useful and friendly to human beings. In this context we started to add to the SLP anticipation ability, which is the first step toward more human-like cooperation ability. Interface between robotic and psychology gave us the fruitful possibility to go further in this direction in [5].

3.1 Anticipation – Extraction behavior regularities

If robots are to engage with humans in useful, timely and cooperative activities, they must be able to learn from their experience, with humans. Ideally this learning should take place in a way that is natural and comfortable for the user. The results of such learning should be that during the course of an interaction, as the robot continuously acquires knowledge of the structure of the interaction, it can apply that knowledge in order to anticipate the behavior of the user. This anticipation can be expressed both in terms of the actions performed by the robot, as well as by its style of verbal communication. Anticipation is the hallmark of cognition: von Hofsten for example says that: Actions are directed to the future and must predict what is going to happen next [26].
We have previously developed cooperation systems that allow a hands-free condition in which the user can actively perform one role in a cooperative task, while instructing the robot at the same time, such that the robot acquired new behaviors via its interaction with the human [2, 3]. The current research thus takes place in the continuity of our studies of human-robot cooperation in the context of a cooperative construction task. The Cooperative Table Construction Task has repetitive subtasks (attaching the 4 legs) which provide an opportunity for learning and performance improvement within the overall task. In previous research, the user was required to explicitly instruct the robot about when to initiate and terminate behavior learning, that is, the user was required to keep track of the segmentation of the overall task into subtasks. The goal, and novelty of the current work, is to extend the learning and anticipation capabilities of the robot within this interactive context by allowing the robot to automatically analyze ongoing behavior with respect to its internal representation of its past. This will allow the robot to anticipate what the user will say (thus improving the spoken language interface), and to take a progressively more proactive role in the interaction. Most importantly, this frees the user from requirement to explicitly segment tasks into subtasks for teaching the robot, as this is now performed automatically.

3.1.1 Multi-level anticipation capability

In order to achieve this anticipatory and adaptive capability we will exploit the notion of the interaction history, as the temporally extended personal sensory motor history of the robot in its interaction with the world and the human [27]. By comparing ongoing interactions with previous experience in the interaction history, the system can begin to anticipate. Depending on the level of stability of these encoded interactions, the system can commit to different levels of anticipatory and initiative-taking behavior. The stability of an interaction will increase as a function of its reuse over time. Here we describe the distinct levels of anticipation and initiative taking that will be implemented and tested. Level 1 anticipation allows the system to predict what the user will say, and thus eliminate the need for verification when the prediction holds. At Level 2 allows the system to take initiative to propose the predicted next event. At Level 3, the robot is highly confident and takes initiative to perform the predicted action.

3.1.1.1 Level 1: Dialog Anticipation

While the speech recognition provided by the CSLU RAD system is quite reliable, we systematically employ a subdialog in which, after the user makes a statement the system asks "Did you say … ?", in order to correct for recognition errors. Most often, the recognition works correctly, and so this verification step is unnecessary, and most of all, it is tiresome for the user.
When the system has recognized that the current sequences of actions matches with a sequence that has been previously executed and stored in the Interaction History, then it can anticipate what will be said next by the user. If this item matches with what is actually recognized as the user's next statement, then the system can dispense with the need to explicitly validate. This can significantly improve the smoothness of the flow of interaction.

3.1.1.2 Level 2: Action Proposition

Once a sequence has been validated at level 1 (i.e. the system correctly predicts what the user will say), then that sequence is elevated to level 2. At this level, again, when the system detects that a level 2 sequence is being executed, it will take initiative and propose to the user the next element in the predicted sequence. The user can then accept or decline the offer. In the context of a repetitive task, this is actually quite helpful as the user can rely on the record of his own previous history with the robot in order to guide ongoing action.

3.1.1.3 Level 3: Action Initiation

Once the sequence has been validated at Level 2, (i.e. the user has accepted the succession of actions proposed by the system), then it attains Level 3. At this level, when the sequence is detected, the robot takes full imitative and begins to execute the subsequent actions in the Level 3 sequence. At this level, the user is truly aided by the "apprentice" who has successively gained confidence, and can now proceed with its part of the interaction without the need to confer by language.

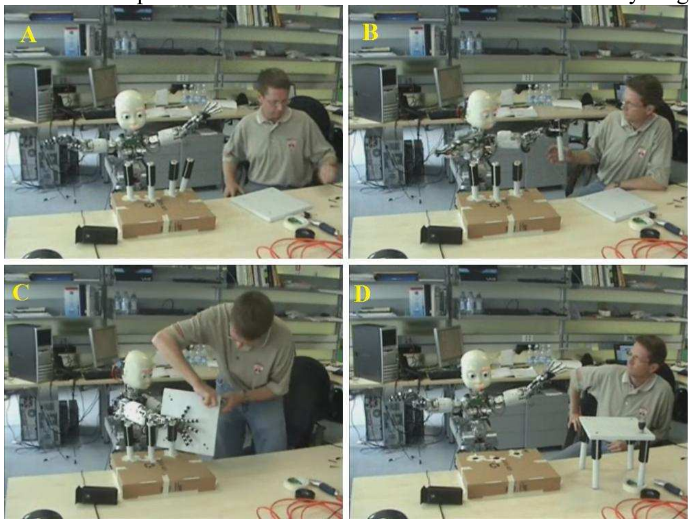

Fig.3. The cooperative table construction task using iCub platform. Robot and human have to cooperate in order to build a table (A,D). The robot has to pass legs to the user (B) and hold the table while the user screws the legs (C).

3.1.1.4 Technical Implementation

The table construction task has been tested using the iCub platform for this research (Fig 3). Part of the system we developed for the current research is illustrated in Fig 4. It describes the control flow that implements the multilevel anticipation capability.

.

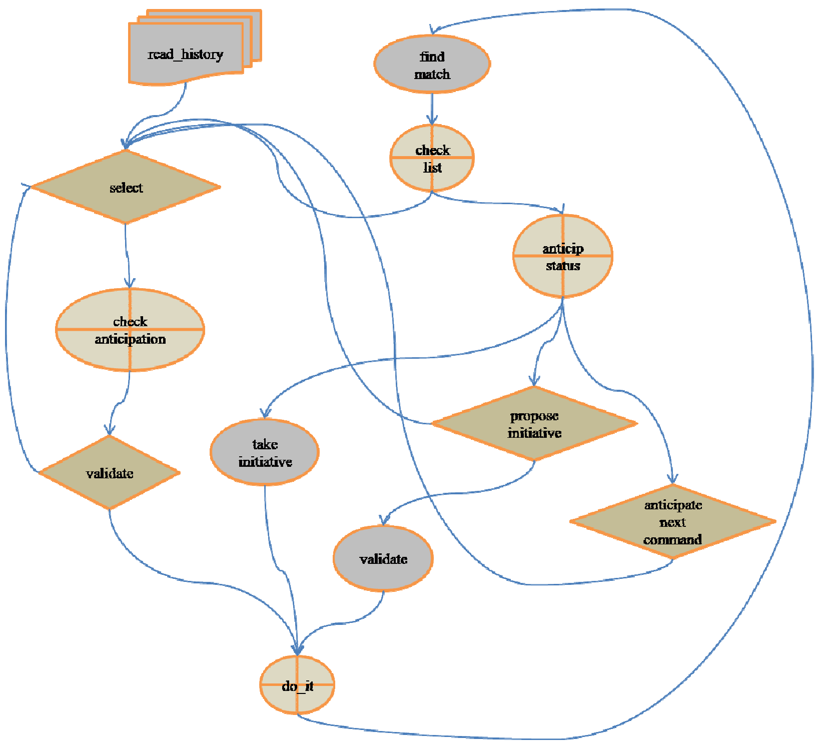

Fig.4. Anticipation's specific part of the SLP. The invariant situation is that the last action commanded has been added to the ongoing Interaction History, and the previous two actions are continuously compared in a sliding window with the Interaction History at *find_match*. If a match is found, the next action in the Interaction History is a candidate for anticipation. At *anticp_status*, if the sequence is recognized for the first time, then at *anticipate_next_command* the next command is identified as a target for speech recognition, and the sequence anticipation level is incremented for the current sequence element. If the sequence is recognized for the second time, at *propose_initiative*, the system proposes this action to the user. If it has been successfully recognized and validated more than twice, the next element is taken for execution at *take_initiative*. At the main node, *Select*, the user chooses actions from the set of atomic actions. *check_anticipation*: Once the action is selected, the system determines what level of anticipation can be applied. If the current action is not part of a sequence of at least two elements recognized in the Interaction History, then there is no anticipation, and the system asks the user to confirm her command. If the command is part of a sequence that has been recognized for the first time – then the system will skip the verbal confirmation if command matches prediction. As already stated, at the 2nd recognition – propose to anticipate, and for 3rd + recognition: take initiative directly

3.1.2 Experimental results

As in the first part of this chapter, we use the Cooperative Table Construction Task to test the system. In this case we just want to test the anticipation ability and not any programming function of the SLP, so we just use a basic set of atomic actions (Reach, Grasp, Lift, Pass, Open, Hold, Wait, Release).

3.1.2.1 Assembling the table

The Cooperative Table Construction Task is interesting as it involves cooperation (the robot must pass elements to the user, and hold things while the user works), and it has a repetitive structure that allows learning. For each leg, the user will ask the robot to reach to and grasp the leg, lift it, pass it to the user and open the hand, and then finally hold the table while the user attaches the leg. In the following interaction, note how the corresponding interaction becomes streamlined and fluid as the robot acquires knowledge of the task.

Reach.
Did you say reach? **Yes** *Reaching*
Grasp.
Did you say grasp? **Yes.** *Grasping.*
Lift.
Did you say lift? **Yes.** *Lifting*
Pass.
Did you say pass? **Yes.***Passing*
Open.
Did you say open? **Yes.** *Opening hand.*
Hold the table.
Did you say hold? **Yes.** *Holding*
Wait. Did you say wait? **Yes.** *Waiting for your signal;*
OK.
Release.
Did you say release? **Yes.** *Releasing*
Wait.
Did you say wait? **Yes.** *Waiting for your signal;*
OK.
Reach.
Did you say reach? **Yes.** *Reaching*
Grasp.
Did you say grasp? **Yes.** *Grasping.*
Lift.
Lifting.
Pass.
Passing
Open.
Opening hand

Hold the table.
Holding
Wait.
Waiting for your signal.
OK
Release.
Releasing.
Reach.
Did you say reach? **Yes.** *Reaching.*
Grasp.
Did you say grasp. **Yes.** *Grasping.*
Shall I do lift? **Yes.** *Lifting.*
Shall I do pass? **Yes.** *Passing.*
Shall I do open? **Yes.** *Opening hand.*
Shall I do hold? **Yes.** *Holding.*
Shall I do wait? **Yes.** *Waiting for your signal.*
OK
Shall I do release? **Yes.** *Releasing.*
Shall I do wait? **No**
Reach.
Did you say reach? **Yes.** *Reaching.*
Grasp.
Grasping.
I know what to do. Lifting.
I know what to do. Passing,
I know what to do. Opening hand,
I know what to do. Holding
I know what to do. Waiting for your signal
OK
I know what to do. Releasing

3.1.2.2 Performance Analysis of SLP Effects

Previous dialog provides evidence that as the levels of anticipation increase, so does the efficiency of the interaction. In order to quantify the effects of this form of anticipatory learning, we measured command processing time as the duration from the spoken onset of the user's command to the robot's completion of that action. We then grouped these actions based on the four assembly phases corresponding to the four legs of the table.
Because of the repetitive structure of the task, the four legs correspond to four levels of anticipation – starting at a naïve state with the first leg at Level 0. Figure 5 presents the mean values and their ranges. We see a reduction in processing time over the course of the task. A repeated measures ANOVA confirmed a significant effect of Level of anticipation on completion time, F(3,15)=4,65; p<,0172.

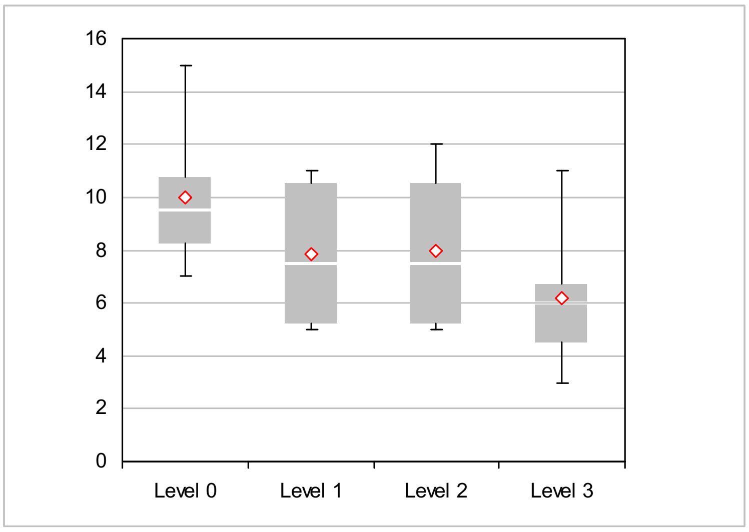

Fig.5. Action execution times (seconds) for robot actions in the assembly of the table, based on level of anticipation. These four levels correspond to the repetitive structure of the task for the 4 legs, respectively. Whiskers indicate max and min times, Boxes 25th and 75th percentile, point – mean, white line – median

3.1.3 Conclusion

The current research focused on the use of the interaction history for allowing the system to automatically identify useful behaviours. The user no longer needs to manage this learning explicitly, but instead can rely on the robot to extract pertinent regularities from their interactions. However, this ability mimics only a small part of the full cooperation capability of a human being. We believe that this global skill should be investigated in order to guide future human robot interaction research.

3.2 Shared plans – Cooperation towards a common goal

There is a fundamental difference between robots that are equipped with sensory, motor and cognitive capabilities, vs. simulations or non-embodied cognitive systems. Via their perceptual and motor capabilities, these robotic systems can interact with humans in an increasingly more "natural" way, physically interacting with shared objects in cooperative action settings. Indeed, such cognitive robotic systems provide a unique opportunity to developmental psychologists for implementing their theories and testing their hypotheses on systems that are becoming increasingly “at home” in the sensory motor and social worlds, where such hypotheses are relevant. The current section reviews our work related to cooperation and learning [5]. It results of interaction between research in computational neuroscience and robotics on the one hand, and developmental psychology on the other. One of the key findings in the developmental psychology context is that with respect to other primates, humans appear to have a unique ability and motivation to share goals and intentions with others. This ability is expressed in cooperative behavior very early in life, and appears to be the basis for subsequent development of social cognition. Here we attempt to identify a set of core functional elements of cooperative behavior and the corresponding shared intentional representations. We then begin to specify how these capabilities can be implemented in a robotic system, the Cooperator, and tested in human-robot interaction experiments. Based on the results of these experiments we discuss the mutual benefit for both fields of the interaction between robotics and developmental psychology.

3.2.1 Introduction

There is a long history of interaction between theoretical aspects of psychology and the information and computer sciences. The “information processing” model of cognitive psychology developed by Neisser [28] and Broadbent [29] borrowed notions such as input, representation, processing and output from computer science and applied them to the analysis of mental processes. Whether or not one holds with specific application of computing metaphors to psychological theories, it appears clear that the use of such metaphors is useful in that it confronts psychological theory with specific questions to be addressed, related to representations and processes underlying cognitive functions. Today the psychological and computing sciences are entering a new period of interaction that is linked to new technological developments in the domain of robotics. Unlike simulation and traditional artificial intelligence programs that are constrained at best to “live” in simulated artificial worlds, robots are equipped with sensory and motor capabilities that allow them to exist in the physical world of the humans that they can interact with. That is, robots can provide experimental platforms to cognitive scientists for implementing and testing theories about the intricate relation between a developing system and its physical environment. Likewise, from the robot technology perspective, robotics scientists have reasoned that the most complex behavior cannot be exclusively programmed by hand, but rather should result from adaptive and developmental mechanisms that are based on those identified in the development of physiological systems [30-32]. One of the most interesting opportunities provided by this interaction between robotics and psychology will be in the domain of developmental psychology. Research in this domain is beginning to focus in on the functional aspects of social cognition that make humans unique in the animal world. It appears that part of the uniquely human aspects concern the ability and motivation to shared intentional states with others [33].The objective of the current research is to begin to identify some of the core elements of the human ability to share intentions based on experimental and theoretical results from developmental psychology, and to then begin to determine how these elements can be implemented on a corresponding robotic system designed for interacting and cooperating with humans. We believe that this work is important because it motivates psychologists to formalize their hypotheses in sufficient detail that they can lead to implementation and testing in artificial but naturally inspired cognitive systems. Of particular interest are the underlying representations required for these shared intentions. We also believe that this work is important because it will begin to endow robots with human-like abilities to cooperate. Tomasello proposed in [33] that the human ability to share intentions develops via the interaction of two distinct capabilities. The first concerns the ability to "read" or determine the intentions of other agents through observation of their behavior, and more generally the ability to represent and understand others as intentional goal directed agents. The second capability concerns the motivation to share intentions with others. While non-human and human primates are skilled at the first - reading the intentions of others based on action and gaze direction, only humans seem to possess an additional capability that will make a significant difference. This is the motivation to cooperate: to share mental states, including goal based intentions which form the basis of cooperation. Perhaps one of the most insightful methods of establishing the properties of human social cognition is the comparison of human and great ape performance in equivalent conditions [34]. In this context, Warneken, Chen and Tomasello [35] engaged 18-to-24 month old children and young chimpanzees in goal-oriented tasks and social games which required cooperation. They were interested both in how the cooperation would proceed under optimal conditions, but also how the children and chimps would respond when the adult stopped performing the task. The principal finding was that children enthusiastically participate both in goal directed cooperative tasks and social games, and spontaneously attempt to reengage and help the adult when he stops. In contrast, chimpanzees are uninterested in non-goal directed social games, and appear wholly fixed on attaining food goals, independent of cooperation. Warneken et al. [36, 37] thus observed what appears to be a very early human capacity for actively engaging in cooperative activities just for the sake of cooperation, and for helping or reengaging the perturbed adult. 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 manipulates 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 reengage the experimenter with a communicative act (on 38% of the interruption trials for 24 month olds), or less often, (b) to attempt to perform the role of the experimenter themselves (on 22% of interruption trials for 24 month olds). Though (b) was considered a non-cooperative behavior, i.e. as an attempt to solve the task individually, it still indicates that the children had a clear awareness both of their role and that of the adult in the shared coordinated activity. Importantly, after only a few demonstrations of the game (and only one demonstration for the 24 month children) it was apparent that the children had a "bird's eye view" or third person representation of the interaction, allowing them to subsequently take either role in the game – that of the launcher or of the receiver of the sliding block. This implies a rather clever representation scheme which can keep track of the goal directed actions of multiple agents, and their interaction, allowing the observer to then take the role of either of the observed agents. In a related study, Warneken & Tomasello [35] demonstrated that 18 and 24 month old children spontaneously help adults in a variety of situations. This is interpreted as evidence for an altruistic motivation to help, and an ability to understand and represent the goals and intentions of others. Indeed, such helping represents a mutual commitment to the shared activity which is one of the defining features of shared cooperative activity [38].The ability to represent the action from multiple perspectives was examined more directly in a study of role reversal imitation conducted by Carpenter et al. [39]. In one experiment of this study, children observed the experimenter cover a "Big Bird" figurine with a cloth. The experimenter then asked the child "Where is big bird? Can you find him?" and the child (or the experimenter) lifted the cloth to reveal the toy. After three such demonstrations, the experimenter handed the cloth to the child and said "It's your turn now." Approximately 70% of the 21 18 month old children tested successfully performed the role reversal. Again, this suggests that the child maintains a representation of the alternating roles of both participants in a third-person perspective that can then be used to allow the child to take on either of the two roles. In order to begin to think about how such a system has come to be (and could be built), we can look to recent results in human and primate neurophysiology and neuroanatomy. It has now become clearly established that neurons in the parietal and the premotor cortices encode simple actions both for the execution of these actions as well as for the perception of these same actions when they performed by a second agent [40, 41]. This research corroborates the emphasis from behavioral studies on the importance of the goal (rather than the details of the means) in action perception [33, 42-44]. It has been suggested that these premotor and parietal "mirror" neurons play a crucial role in imitation, as they provide a common representation for the perception and subsequent execution of a given action. Interestingly, however, it has been clearly demonstrated that the imitation ability of non-human primates is severely impoverished when compared to that of humans [33, 41]. This indicates that the human ability to imitate novel actions and action sequences in real time (i.e. after only one or two demonstrations) relies on additional neural mechanisms to those found in non-human primates. In this context, a recent study of human imitation learning [45] implicates Brodmann's area (BA) 46 as responsible for orchestrating and selecting the appropriate actions in novel imitation tasks. We have recently proposed that BA 46 participates in a dorsal stream mechanism for the manipulation of variables in abstract sequences and language [46]. Thus, variable "slots" that can be instantiated by arbitrary motor primitives during the observation of new behavior sequences are controlled in BA 46, and their sequential structure is under the control of corticostriatal systems which have been clearly implicated in sensorimotor sequencing [46]. This allows us to propose that this evolutionarily more recent cortical area BA 46 may play a crucial role in allowing humans to perform compositional operations (i.e. sequence learning) on more primitive action representations in the ventral premotor and parietal motor cortices. In other words, ventral premotor and parietal cortices instantiate shared perceptual and motor representations of atomic actions, and BA46 provides the capability to compose arbitrary sequences of these atomic actions, while relying on well known corticostriatal neurophysiology for sequence storage and retrieval. The functional result is the human ability to observe and represent novel behavioral action sequences. We further claim that this system can represent behavioral sequences from the "bird's eye view" or third person perspective, as required for the cooperative tasks of Warneken et al. [36]. That is, it can allow one observer to perceive and form an integrated representation of the coordinated actions of two other agents engaged in a cooperative activity. The observer can then use this representation to step in and play the role of either of the two agents. This is a "dialogic cognitive representation," or "we intention" in that it represents the "dialog" of interaction between agents. Given this overview of some of the core functional elements of cooperative behavior and the corresponding representations (including the "bird's eye view"), we can now take on the task of beginning to specify how these capabilities can be implemented in a robotic system, and tested in human-robot interaction experiments. When making the transition from human behaviour to technological implantation, there is the risk that the implementation will be biased in terms of specific computational or functionalist solutions. In this context, we are making a concerted effort in "cognitive systems engineering," a process in which the cognitive robotics systems we build are constrained by (1) functional requirements (i.e. specification of how the system behaves) derived from behavior from developmental psychology, and (2) architectural constraints from the neurosciences. To as large a degree as possible, we avoid arbitrary constraints from the purely computational aspects of the implementation platform.

3.2.2 The robotic system – The Cooperator

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 cooperative activity will involve interactive tasks that preserve the important aspects of the "block launching" task of Warneken et al. [36], transposed into a domain of objects suitable for our robot system. 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 Fig 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. Fig 6 provides an overview of the architecture, and Fig 7, which corresponds to Experiment 6 provides an overview of the actual physical state of affairs during a cooperative interaction.

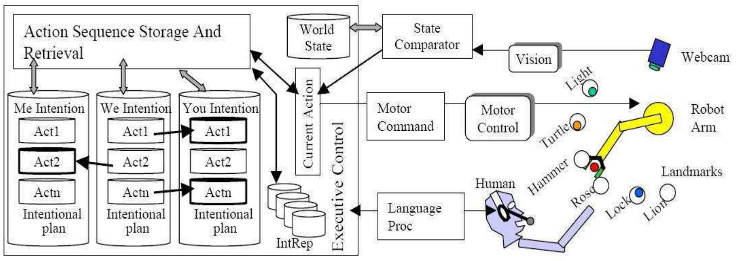

Fig.6. Cooperation System. In a shared workspace, human and robot manipulate objects (green, yellow, red and blue circles corresponding to dog, horse, pig and duck), placing them next to the fixed landmarks (light, turtle, hammer, etc.). *Action*: Spoken commands interpreted as individual words or grammatical constructions, and the command and possible arguments are extracted using grammatical constructions in Language Proc. The resulting Action(Agent,Object,Recipient) representation is 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 the 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.

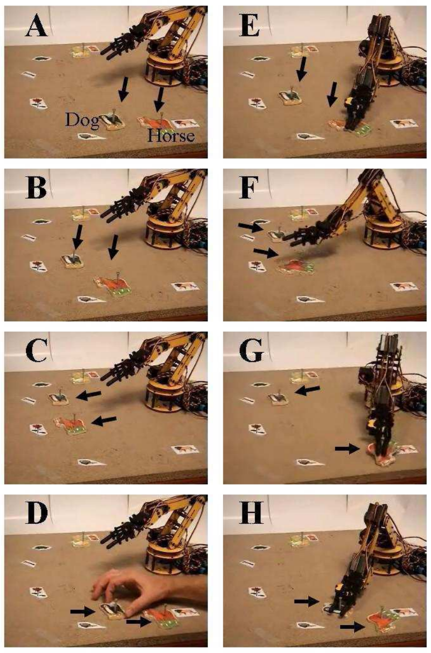

Fig.7. Cooperative task of Exp 5-6. Robot arm Cooperator, with 6 landmarks (Light, turtle, hammer, rose, lock and lion from top to bottom). Moveable objects include Dog and Horse. In A-D, human demonstrates a “horse chase the dog” game, and successively moves the Dog then Horse, indicating that in the 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.

3.3.2.1 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. In the context of the current study, actions involve moving objects to different locations, and are identified by the agent, the object, and the target location the object is moved to. As illustrated in Fig 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-based action imitation. This will be expanded upon below. In order to allow for more elaborate cooperative activity, the system must be able to store and retrieve actions in a sequential structure, and must be able to associate each action with its agent. We thus propose that the ability to store a sequence of actions, each tagged with its agent, provides an initial capability for dialogic cognitive representation. This form of real time sequence learning for imitation is not observed in non-human primates[41]. In this context, an fMRI study [45] which addressed the human ability to observe and program arbitrary actions indicated that a cortical area (BA46) which is of relatively recent phylogenetic origin is involved in such processes. Rizzolatti and Craighero [41] have thus suggested that the BA 46 in man will orchestrate the real-time capability to store and retrieve recognized actions, and we can further propose that this orchestration will recruit canonical brain circuitry for sequence processing including the cortico-striatal system (see [46, 47] for discussion of such sequence processing). An additional important representational 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 and Roy [14]. The World Model encodes the physical locations of objects and is updated by vision and proprioception (i.e. robot action updates World Model with new object location). Changes observed in the World Model in terms of an object being moved allows the system to detect actions in terms of 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 the current study we address behavioral conditions which focus on the observation and immediate re-use of an intentional (goal directed) action plan. However, in the more general case, one should consider that multiple intentional action plans can be observed and stored in a repertory (IntRep or Intentional Plan Repertory in Fig 7). When the system is subsequently observing the behavior of others, it can compare the ongoing behavior to these stored sequences. Detection of a match with the beginning of a stored sequence can be used to retrieve the entire sequence. This can then be used to allow the system to "jump into" the scenario, to anticipate the other agent's actions, and/or to help that agent if there is a problem.

3.3.2.2 Cooperation Control Architecture

As in the SLP, the spoken language control architecture illustrated in Fig 8 is implemented with the CSLU RAD. 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). Most importantly 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. Fig 8 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 via spoken commands (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 [8-11, 46] is used to extract the action that the robot then performs, in the form *Move(object, location)*. 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 then begins to visually observe the scene until it detects the action, based on changes in object locations 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. The more general case corresponds to “games” in which the robot and human will take turns in the execution of a shared plan. In the current implementation of this, the user can demonstrate multiple successive actions, and indicate the agent (by saying “You/I do this”) for each action. Improvements in the visual processing will allow the more general case in which the system can observe two agents interacting and attribute each action to its respective agent. 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 that action 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. Interestingly, the ability of the robot to “help” the user comes quite naturally, based on the shared intentional plan. If the user action is not performed, the robot “knows” the failed action based on its own representation of the plan. The robot can thus communicate with the user, and if the user agrees, the robot can help by performing the action itself. Thus, “helping” was quite naturally implemented by combining an evaluation of the user action, with the existing capability to perform a stored action representation. Still, it is worth noting that one crucial difference between the helping by the robot and what Warneken et al. tested in the helping study [35] was that the children and chimpanzees helped the other *with* their action, not just performing the other’s action completely, but complementing the other’s action.

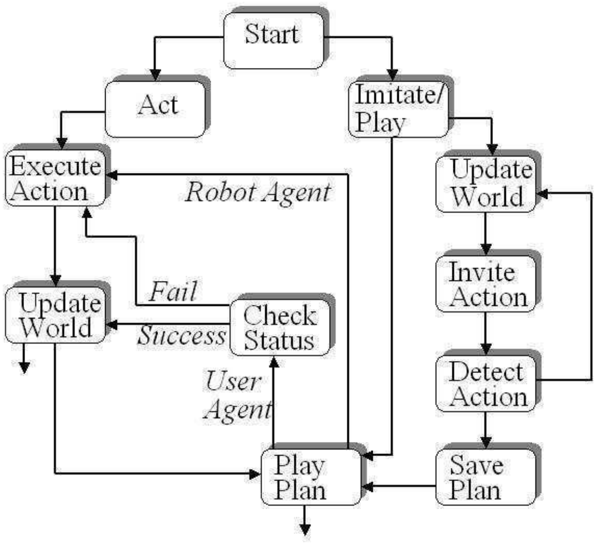

Fig.8. Spoken Language Based Cooperation flow of control. Interaction begins with proposal to act, or imitate/play a game. Act – user says an action that is verified and executed by robot. World Model updated based on action. Downward arrow indicates return to Start. Imitate/Play – user demonstrates actions to robot and says who the agent should be when the game is to be played (e.g.“You/I do this”). Each time, system checks the state of the world, invites the next action and detects the action based on visual object movement. When the demo is finished, the plan (of a single item in the case of imitation) is stored and executed (Play Plan). If the user is the agent (encoded as part of the game sequence), system checks execution status and helps user if failure. If robot is agent, system executes action, and then moves on to next item.

3.3.2.3 Bird’s Eye View and Role Reversal

In an initial set of experiments (Experiments 1-6 below), the “intentional plan” was represented for the robot as a sequence of actions in the “We Intention” of Fig 6, with the attribution of the agent fixed for each action. We know however from the experimental results of Warneken et al. [36], and from the role reversal studies of [39] that this representation is flexible, in the sense that the child can take on the role of either of the two represented agents. Once the adult indicates the role he takes, the child then spontaneously adapts and takes the other role. In the current system, we thus introduce a new capability in which, prior to the playing of the game, the roles can be determined and modified. When control reaches the “Plan Play” node in the controller, i.e. after a new game has been demonstrated, or after the user chooses to play the old game, the robot now asks the user if he wants to go first. If the user responds yes, then the roles of user and robot remain as they were in the demonstration. If the user says no, then the roles are reversed. Reversal corresponds to systematically reassigning the agents (i.e. robot or user) associated with each action. Indeed, technically it would be possible that based upon the first move by the user (or the users insistent waiting for the robot to start), the robot infers who does what (i.e. whether to reverse roles or not) and what role it will take in the cooperative plan, though this has was not implemented in the current version of the system.

3.2.3 Experimental Results

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 overall, 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.

3.2.3.1 Experiment 1: Validation of Sensorimotor Control

In this experiment, the user says that he wants the "Act" state (Fig 6), 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 objects to and from different landmark locations. In these ten experimental runs, the system performed correctly. Experiment 1 thus demonstrates that the system has (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.

3.2.3.2 Experiment 2: 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 detailed trajectory or means by which these ends were achieved [39, 42]. 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 6) 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 6, and Detect Action in Fig 8, 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). In ten experimental runs the system performed correctly. This demonstrates the ability of the system to detect the final “goal” of user-generated actions as defined by visually perceived state changes, and the utility of a common representation of action for perception, description and execution.

3.2.3.3 Experiment 3: 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, turn taking 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. [36] the robot invites the user to start showing how the game works. Note that in these experiments, two experimenters demonstrate the game and the subject is observing this interaction from a third-person-perspective. The experimenters invite the child to see how the game works by showing it to them first and then have them participate afterwards. For technical limitations of the visual system, we currently use the following modification: The user then begins to perform a sequence of actions that are observed by the robot. 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. Note that because the system can detect actions, if it is capable of identifying distinct users (by some visual cue on their hands for example) then the system could observe two humans perform the task, thus adhering more closely to the protocol of Warneken et al. [36]. 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.

3.2.3.4 Experiment 4: Interrupting a Cooperative Game

In this experiment, everything proceeds as in experiment 3, 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.

3.2.3.5 Experiment 5: A More Complex Game

Experiment 3 represented the simplest behavior that could qualify as a cooperative action sequence. In order to more explicitly test the intentional sequencing capability of the system, this experiment replicates Exp 3 but with a more complex task. In this game, the user starts by moving the dog, and after each move the robot “chases” the dog with the horse, until they both return to their starting places. Action User identifies agent User Demonstrates Action Ref in Figure 6 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 1. 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 Fig 6. As in Experiment 3, 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 in Exp 3, two different complex games were learned, and each one “played” successfully 5 times. This illustrates the learning by demonstration [15] of a complex intentional plan in which the human and the robot are agents in a coordinated and cooperative activity.

3.2.3.6 Experiment 6: 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 demonstrates the generalized ability to help that can occur whenever the robot detects the user is in trouble.

3.2.3.7 Experiment 7: Role reversal in the Complex Game

Carpenter et al. [43] demonstrated that 18 month old children can observe and participate in a cooperative turn-taking task, and then reverse their role, indicating that they develop a third person “bird’s eye view” perspective of the interaction. The current experiment tests the ability of the system to benefit from the “bird’s eye view” representation of the shared intentional plan in order to take either role in the plan. In one test, the same “old game” from experiments 5 and 6 was used, with the modified version of the system that asks, prior to playing the game “do you want to go first”. To test the role reversal, the human responds “no”. In the demonstrated game, the human went first, so the “no” response constitutes a role reversal. The system thus systematically reassigns the You and Me actions of the We intention in Fig 6. Once this reassignment has been made, then the shared plan execution mechanism proceeds in the standard manner. The system successfully performed this role reversal. Again, it is technically feasible for the robot to infer its own role based upon only what the user does, by detecting whether or not the user initiates the first action in the game, and such an implementation will be pursued in our future work.

3.3 Discussion

Significant progress has been made in identifying some of the fundamental characteristics of human cognition in the context of cooperative interaction, particularly with respect to social cognition [48-51]. Breazeal and Scassellati [52] investigate how perception of socially relevant face stimuli and object motion will both influence the emotional and attentional state of the system and thus the human-robot interaction. Scassellati [53] further investigates how developmental theories of human social cognition can be implemented in robots. In this context, Kozima and Yano [50] outline how a robot can attain intentionality – the linking of goal states with intentional actions to achieve those goals – based on innate capabilities including: sensory-motor function and a simple behavior repertoire, drives, an evaluation function, and a learning mechanism. 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. 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 old children [35, 36]. While the primitive basis of such behavior is visible in chimpanzees, its full expression is uniquely human. As such, it can be considered a crucial component of human-like behavior for robots. 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 [7-11, 46, 47]. 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 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 [33, 47]. In this context, we are currently investigating the development of hierarchical means-end action sequences [44]. At each step, the objective will be to identify the characteristic underlying behavior and to implement it in the most economic manner in this continuously developing system for human robot cooperation. Here we begin to address the mechanisms that allow agents to make 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 18-24 months, the child can thus observe the two roles being played out, and then step into either role [43]. 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, and associates one of these with 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 is consistent with the system illustrated in Fig 6. We could improve the system: 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. In the current system we demonstrate that the roles associated with “you” and “me” can be reversed. More generally, they can also be dissociated from “you” and “me” and linked with other agents. The key is that there is a central representation corresponding to the “We intention” in Figure 6, which allows the “bird’s eye view”, and a remapping mechanism that can then assign these component actions to their respective agents (corresponding to the Me and You intentions in Fig 6). Clearly there remains work to be done in this area, but the current results represent a first step in specifying how these intentional representations could be implemented. Indeed, we take a clear position in terms of internal representational requirements, defined by a hybrid form of representation. At one level, online action and perception are encoded in an “embodied” form in terms of joint angles, and continuous values from the visual system. At a different level, “we intentions” which allow an extension in time, are distinct sequences of predicate-argument propositional elements. Thus there is a continuum of embodiment and representation. In the context of representing a joint activity through observation – the action perception is linked to the sensorimotor system, but the system that stores and replays these sequences can be considered to be independent. Indeed, it is this simulation capability that might well provide the basis for abstract processing [54]. More broadly speaking, though the demands of requiring implementation, robot experiments such as these can help us to shed further light on the nature and necessity of internal representations. An important open issue that has arisen through this research 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. Indeed, this development in the robotics context provides interesting predictions about how these inferences will be made that can be tested with children. A potential criticism of this work could hold that while it might demonstrate an interesting and sophisticated simulation, everything of interest seems to be built in rather than emergent or developed, thus of relatively thin relevance to psychologists. We would respond that any implementation must make choices about what is built in and what is emergent. Here we have built in functions that provide the ability to perceive actions, encode action-agent sequences, and to use these sequences in behaviour. What results is the open ended capability to learn arbitrary cooperative behaviors, to help, and to changes perspectives/roles. The relevance to psychologists is twofold, in terms of what the resulting system can do, and in terms of where it fails. Thus, while we have begun to implement some aspects of these intention representations, we should also stress how the robot's capabilities still differ from what these young children already do, including the following. (1) Children learn intentional plans quickly without direct teaching, but just by observing from the outside how two people interact. (2) They are not told who performs which role, but they themselves are able to parse the interaction into roles. (3) They spontaneously provide help without the experimenter asking them for help and without them asking the experimenter whether he wants help. (4) They not only help the other with his role but they insist on the partner performing his role when he interrupts. In other words, they seem to insist on the joint commitment to perform the respective roles. For the most part, these differences are "peripheral" in that they are related to the perception and action capabilities, rather than to the structure of internal representations. Point (1) will rely on a "salience" system that determines what behavior is interesting and merits learning (perhaps any behavior between multiple agents operating on the same objects). Point (2) will require vision processing that allows identification of different individuals. For points (3) and (4), the behavior is currently available, i.e. it is wholly feasible for the robot to help and to insist that the other partner participates spontaneously as the situation requires. In conclusion, the current research has attempted to build and test the Cooperator, a robotic system for cooperative 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 and flexible alternation by the human and robot. While many technical aspects of robotics (including visuomotor coordination and vision) have been simplified, we believe that this work makes a useful contribution in demonstrating how empirical and theoretical results in developmental psychology can be formalized to the extent that they can be implemented and tested in a robotic system. In doing so, we gain further insight into the core functions required for cooperation, and help to increase the cooperative capabilities of robots in human-robot interaction.

4 Conclusion

In the future, robots will cooperate with human beings in everyday life. Even (and especially) people that are not involved in any kind of programming or computer science should be able to command, ask for help or play with a robot. In this context, robots will have to adapt to human desires through social interaction, which include spoken language, behavior and intention recognition, etc. In this chapter, we reviewed our work in this direction. We believe that the most natural modality for social interaction is spoken language, which led us to build the Spoken Language Programming system. This tool provides a way to control a robot efficiently, but the lack of free will and intentionality of the robot is somehow restrictive and result in an interaction that is too artificial and rigid. Cooperative abilities like the anticipation of actions or the will to help people have to be embedded in the SLP in order to build a complete robotic collaborative system. Such skills will elevate the robot behaviors to a new level of fluency, allowing the robot to become a partner instead of just being commanded. Our studies have already extracted some of these skills from human interaction based experiments and we begin to implement them within the context of human-robot cooperation. Such collaborations between domains of psychology and robotic are still novel and will lead to most interesting results in the future.

5 Acknowledgements

This work has been supported in part by the European FP7 ICT project CHRIS, and by the French ANR Amorces.

6 References

1. Crangle, C., Suppes, P. Language and Learning for Robots. in CSLI lecture notes. 1994. Stanford.
2. Dominey, P., Mallet, A, Yoshida, E. Progress in Programming the HRP-2 Humanoid Using spoken Language. in ICRA. 2007.
3. Dominey, P., Mallet, A, Yoshida, E. Real-Time Cooperative Behavior Acquisition by a Humanoid Apprentice. in IEEE/RAS International Conference on Humanoid Robotics. 2007. Pittsburg Pennsylvania.
4. Dominey, P., Metta, G., Nori, F., Natale, L. Anticipation and Initiative in Human-Humanoid Interaction. in Humanoids. 2008.
5. Dominey, P., Warneken, F, The Basis of Shared Intentions in Human and Robot Cognition. In press, 2008.
6. Goldberg, A., Constructions: A new theoretical approach to language. Trends in Cognitive Sciences, 2003. **7**: p. 219–24.
7. Dominey, P., Spoken Language and Vision for Adaptive Human-Robot Cooperation, in Humanoid Robotics ARS International. 2007, Matthias Hackel, Ed: Vienna.
8. Dominey, P., Boucher, JD, Learning To Talk About Events From Narrated Video in the Construction Grammar Framework. Artificial Intelligence, 2005. **167**: p. 31–61.
9. Dominey, P. Learning grammatical constructions from narrated video events for human–robot interaction. in Proceedings IEEE Humanoid Robotics Conference. 2003. Karlsruhe, Germany.

10. Dominey, P., Boucher, JD, Inui, T Building an adaptive spoken language interface for perceptually grounded human–robot interaction. in Proceedings of the IEEE-RAS/RSJ international conference on humanoid robots. 2004.
11. Dominey, P., Alvarez, M, Gao, B, Jeambrun, M, Weitzenfeld, A, Medrano, A Robot Command, Interrogation and Teaching via Social Interaction. in IEEE Conference On Humanoid Robotics. 2005.
12. Nicolescu, M., Mataric, MJ. Learning and Interacting in Human-Robot Domains. in IEEE Trans. Sys. Man Cybernetics.
13. Lauria S, B.G., Kyriacou T, Klein E, Mobile robot programming using natural language. Robotics and Autonomous Systems, 2002. **38(3-4)**: p. 171-181.
14. Mavridis, N., Roy, D. Grounded Situation Models for Robots:Where Words and Percepts Meet. in Proceedings of the IEEE/RSJInternational Conference on Intelligent Robots and Systems (IROS). 2006.
15. Zöllner, R., Asfour, T, Dillman, R. Programming by Demonstration:Dual-Arm Manipulation Tasks for Humanoid Robots. in Proc IEEE/RSJ Intern. Confon Intelligent Robots and systems (IROS). 2004.
16. Thorpe, S., A. Delorme, and R. Van Rullen, Spike-based strategies for rapid processing. Neural Netw, 2001. **14**(6-7): p. 715-25.
17. Kaneko, K., Kanehiro, F, Kajita, S, Hirukawa, H, Kawasaki, T, Hirata, M, Akachi, K, Isozumi, T. Humanoid robot hrp-2. in EEE International Conference on Robotics & Automation. 2004.
18. Kanehiro, F., Miyata, N, Kajita, S, Fujiwara, K, Hirukawa, H, Nakamura, Y, Yamane, K, Kohara, I, Kawamura, Y, Sankai, Y. Virtual Humanoid Robot Platform to Develop Controllers of Real Humanoid Robots without Porting. in Int. Conference on Intelligent Robots and Systems. 2001.
19. Tsagarakis, N.G., Metta, G, Sandini, G, Vernon, D, Beira, R, Becchi, F, RIghetti, L, Victor, J.S, Ijspeert, A.J, Carrozza, M.C, Caldwell, D.G, iCub - The Design and Realization of an Open Humanoid Platform for Cognitive and Neuroscience Reseach. Advanced Robotics, 2007. **21**(Robotic platforms for Research in Neuroscience).
20. Fitzpatrick, P., Metta, G, Natale, L, Towards Long-Lived Robot Genes. Journal of Robotics and Autonomous System, 2008. **56**(Special Issue on Humanoid Technologies): p. 1-3.
21. Severinson-Eklund, K., Green, A, Hüttenrauch, H, Social and collaborative aspects of interaction with a service robot. Robotics and Autonomous Systems, 2003. **42**: p. 223–234.
22. Calinon, S., Guenter, F, Billard, A. On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts. in IEEE/ICRA 2006. 2006.
23. Kyriacou, T., Bugmann, G, Lauria, S, Vision-based urbannavigation procedures for verbally instructed robots. Robotics andAutonomous Systems, 2005. **51**: p. 69-80.
24. Delorme, A. and S.J. Thorpe, SpikeNET: an event-driven simulation package for modelling large networks of spiking neurons. Network, 2003. **14**(4): p. 613-27.
25. Turing, A.M. Computing Machinery and Intelligence. Available from: http://www.abelard.org/turpap/turpap.htm.
26. von Hofsten, C., An action perspective on motor development. Trends in Cognitive Sciences, 2004. **8**(6): p. 266-272.
27. Mirza, N., Nehaniv, CL, Dautenhahn, K, Boekhorst, R, Grounded Sensorimotor Interaction Histories in an Information THeoretic Metric Space for Robot Ontogony. Adaptive Behavior, 2007. **15**: p. 167-187.
28. Neisser, U., Cognitive psychology Appleton-Century-Crofts 1967: New York.
29. Broadbent, D.E., Information processing in the nervous system. Science, 1965. **150**(695): p. 457-62.
30. Brooks, R., Elephants Don't Play Chess. Robotics and Autonomous Systems, 1990. **6(3)**: p. 3-15.
31. Pfeifer, R., Scheier, C, Understanding Intelligence. 1999, Cambridge: The MIT Press.
32. Pfeifer, R., Gómez, G, Interacting with the real world: design principles for intelligentsystems. Artificial Life and Robotics, 2005. **9**(1): p. 1-6.

33. Tomasello, M., et al., Understanding and sharing intentions: the origins of cultural cognition. Behav Brain Sci, 2005. **28**(5): p. 675-91; discussion 691-735.
34. Tomasello, M. and M. Carpenter, Shared intentionality. Dev Sci, 2007. **10**(1): p. 121-5.
35. Warneken, F. and M. Tomasello, Altruistic helping in human infants and young chimpanzees. Science, 2006. **311**(5765): p. 1301-3.
36. Warneken, F., F. Chen, and M. Tomasello, Cooperative activities in young children and chimpanzees. Child Dev, 2006. **77**(3): p. 640-63.
37. Warneken, F., et al., Spontaneous Altruism by Chimpanzees and Young Children. PLoS Biol, 2007. **5**(7): p. e184.
38. Bratman, M., Shared cooperative activity. The Philosophical Review, 1992. **101**(2): p. 327-341.
39. Carpenter, M., Tomasello, M, Striano, T, Role reversal imitation and language in typically-developing infants and children with Autism. Infancy, 2005. **8(3)**: p. 253-287.
40. di Pellegrino, G., et al., Understanding motor events: a neurophysiological study. Exp Brain Res, 1992. **91**(1): p. 176-80.
41. Rizzolatti, G. and L. Craighero, The mirror-neuron system. Annu Rev Neurosci, 2004. **27**: p. 169-92.
42. Bekkering, H., A. Wohlschlager, and M. Gattis, Imitation of gestures in children is goal-directed. Q J Exp Psychol A, 2000. **53**(1): p. 153-64.
43. Carpenter, M., Call, J, The question of 'what to imitate': inferring goals and intentions from demonstrations, in Imitation and Social Learning in Robots, Human sand Animals, C.L.N.a.K.D. Eds, Editor. 2007, Cambridge University Press: Cambridge.
44. Sommerville, J.A. and A.L. Woodward, Pulling out the intentional structure of action: the relation between action processing and action production in infancy. Cognition, 2005. **95**(1): p. 1-30.
45. Buchine, G., Vogt, S, Ritzl, A, Fink, GR, Zilles, K, Freund, H-J, Rizzolatti, G, Neural circuits Underlying Imitation Learning of Hand Actions: An Event-Related fMRI Study. Neuron, 2004. **42**: p. 323-334.
46. Dominey, P.F., M. Hoen, and T. Inui, A neurolinguistic model of grammatical construction processing. J Cogn Neurosci, 2006. **18**(12): p. 2088-107.
47. Dominey, P., Toward a construction-based account of shared intentions in social cognition. Behav Brain Sci, 2005. **28:5**: p. 696.
48. Fong, T., Nourbakhsh, I, Dautenhaln, K, A survey of socially interactive robots. Robotics and Autonomous Systems, 2003. **42**(3-4): p. 143-166.
49. Goga, I., Billard, A, Development of goal-directed imitation, object manipulation and language in humans and robots, in Action to Language via the Mirror Neuron System, M.A.A. (ed.), Editor. 2005, Cambridge University Press: Cambridge.
50. Kozima H, Y.H. A robot that learns to communicate with human caregivers. in International Workshop on Epigenetic Robotics. 2001.
51. Lieberman, M.D., Social cognitive neuroscience: a review of core processes. Annu Rev Psychol, 2007. **58**: p. 259-89.
52. Breazeal, C., Scassellati, B,, Challenges in building robots that imitate people. K. Dautenhahn, 2001.
53. Scassellati, B., Theory of mind for a humanoid robot. Autonomous Robots, 2002. **12**(1): p. 13-24.
54. Barsalou, L.W., Perceptual symbol systems. Behav Brain Sci, 1999. **22**(4): p. 577-609; discussion 610-60.