Preprint version; final version available at http://ieeexplore.ieee.org
Transactions on Autonomous Mental Development (2013), vol: 5(1), pp: 3-17
Published by: IEEE
DOI: 10.1109/TAMD.2012.2209880

The Coordinating Role of Language in Real-Time Multi-Modal Learning of Cooperative Tasks

Maxime Petit[1], Stéphane Lallée[1], Jean-David Boucher[1], Grégoire Pointeau[1], Pierrick Cheminade[1], Dimitri Ognibene[2], Eris Chinellato[2], Ugo Pattacini[3], Ilaria Gori[3], Uriel Martinez-Hernandez[4], Hector Barron-Gonzalez[4], Martin Inderbitzin[5], Andre Luvizotto[5], Vicky Vouloutsi[5], Yannis Demiris[2], Giorgio Metta[3] and Peter Ford Dominey[1]

[1] INSERM U846 SBRI, Robot Cognition Laboratory, 18 avenue Doyen Lepine, 69675 Bron, France: [2] Imperial College of Science, Technology and Medicine, London, UK; [3] Fondazione Istituto Italiano di Tecnologia, Genoa, Italy; [4] University of Sheffield, UK, [5] Universitat Pompeu Fabra, Barcelona, Spain:

(maxime.petit, peter.dominey)@inserm.fr

One of the defining characteristics of human cognition is our outstanding capacity to cooperate. A central requirement for cooperation is the ability to establish a "shared plan" – which defines the interlaced actions of the two cooperating agents – in real time, and even to negotiate this shared plan during its execution. In the current research we identify the requirements for cooperation, extending our earlier work in this area. These requirements include the ability to negotiate a shared plan using spoken language, to learn new component actions within that plan, based on visual observation and kinesthetic demonstration, and finally to coordinate all of these functions in real time. We present a cognitive system that implements these requirements, and demonstrate the system's ability to allow a Nao humanoid robot to learn a non-trivial cooperative task in real-time. We further provide a concrete demonstration of how the real-time learning capability can be easily deployed on different platform, in this case the iCub humanoid. The results are considered in the context of how the development of language in the human infant provides a powerful lever in the development of cooperative plans from lower-level sensorimotor capabilities.

***Index Terms*—cooperation, humanoid robot, spoken language interaction, shared plans, situated and social learning.**

I. Introduction

THE ability to cooperate, to creatively establish and use shared action plans is, like language and the underlying social cognitive and motivational infrastructure of communication, one of the major cognitive capacities that separates humans from non-human primates [1]. In this context, language itself is an inherently cooperative activity in which the listener and speaker cooperate, in order to arrive at the shared goal of communication. Tomasello et al make the foundational statement that language is built on the uniquely human ability to read and share intentions, which is also the foundation for the uniquely human ability and motivation to cooperate. Indeed, Tomasello goes one step further, suggesting that the principal function of language is to establish and negotiate cooperative plans [1].

The building blocks of cooperative plans are actions. In this context, it has been suggested that we are born with certain systems of "core cognition", which are "identified by modular innate perceptual-input devices" [2] (p. 11). One of the proposed elements of core cognition is agency. This includes an innate system for representing others in terms of their goal directed actions, and perceptual mechanisms such as gaze following that allow the developing child to monitor the goal directed actions of others. Thus we consider that these notions of agency are given in the system, though the degree to which they may actually be developed vs. innate remains an open question [2].

A cooperative plan (or shared plan) is defined as a goal directed action plan, consisting of interlaced turn-taking actions by two cooperating agents, in order to achieve a common goal that could otherwise not have been achieved individually [1]. Interestingly, infants can establish shared plans without the use of language, if the shared goal and corresponding plan are sufficiently simple. However, once the plans reach a certain level of complexity, and particularly if the plan must be renegotiated in real-time, then language is often invoked to establish and negotiate who does what [3, 4]. Thus, cooperation requires communication, and when things get complex, language is the preferred communication method. Indeed, much of early language maps onto physical parameters of goal directed action [5, 6].

In the construction grammar framework, Goldberg identifies how the structure of language is mapped onto the structure of meaning such that "constructions involving basic argument structure are shown to be associated with dynamic scenes … such as that of someone volitionally transferring something to someone else, someone causing something to move or change state" [5] p. 5. Thus, grammatical constructions implement the mapping from linguistic utterances to meaning, in the form of action and perceptual scene specifications. The nature of the link between language and action, and how that link is established, is an open topic of research in child development and developmental robotics [7].

In the context of this debate, following a usage-based approach [6], we have demonstrated how such constructions can be learned in a usage-based approach, as the mapping between the argument structure of sentences and argument structure of robotic representations of action meanings [8]. This "usage-based" development of grammatical constructions (vs. a more nativist approach) is also a topic of debate, similar to the case for agency cited above.

Independent of the nativist vs. usage-based debate, we can take the position that via such constructions, language is

uniquely situated in its capability to allow agents to construct and negotiate shared cooperative plans. Our approach is to implement a scaffolded system based on this capability. In this scaffolding, we build in simple grammatical constructions that map onto the argument structure of actions that can be performed by the robot. This allows a scaffolding for the creation of action plans. We have previously used spoken language to construct diverse action plans for a robot cooperating with a human [9, 10], but the plans were not shared, in that they only specified the robot's actions. We then introduced a shared planning capability where a robot could observe a sequence of actions, with an agent attributed to each by the user via language. This generated a true shared plan, that could pass the test of role reversal [11]. Role reversal occurs when the two participants in a cooperative task can exchange roles, thus indicating that they both have a "bird's eye view" of the shared plan, which is a central part of the requirements for true cooperation [12].

In a series of studies we then more carefully re-examined the bases of shared planning. In the first study [13] we implemented a capability for learning to perceive and recognize novel human actions based on the structure of perceptual primitive constituting those actions. We next implemented the corresponding ability to learn to execute complex actions based on the composition of motor primitives, and to make the link between perception and action via imitation [14]. Finally, we extended this capability to multiple actions in shared plans, where the human could use spoken language to specify a shared plan that could then be executed by the robot, again displaying role reversal [15].

While this work represented significant progress, it left several issues unanswered. First, when a shared plan "goes wrong" there is no mechanism to fix it. Language can fulfill this role -indeed much of human language is about coordinating and correcting shared plans [16]. Second, in our previous work, teaching the shared plan was in a fixed modality, typically with the human speaking the shared plan, action by action. Here we extend this so that language becomes the central coordinator, a scaffold, which allows the user to then specify individual actions by (a) kinesthetically demonstrating the action, (b) performing the action himself so the robot can perceive and imitate, or (c) finally - for known actions – to specify the action verbally. Learning by visual and kinesthetic demonstration are highly developed and well documented means for transmission of skill from human to robot e.g. [17-19]. We will demonstrate how this provides a novel interaction framework that where language coordinates these three potential modalities for learning shared plans.

The transmission of knowledge from humans to robots can take multiple forms. We consider three specific forms. "Imitation" will refer to learning in which the human performs the action to be learned, and the robot observes this and performs a mapping from observation space onto its execution space, as defined in [20]. Likewise based on [20] we will refer to "kinesthetic teaching" as a form of "demonstration" where the passive robot is moved through the desired trajectory by the human teacher. Finally we will refer to "spoken language programming" [21] as the method described above where well-formed sentences are used to specific robot actions and arguments, either in isolation or in structured sequences. Language has been used to explain new tasks to robots [22], and is especially useful for scaffolding tasks, when the teacher uses previously acquired skills to resolve a new and more complex tasks [23].

Imitation has been successfully used on diverse platforms [24-29]. It is an easy way for the teacher to give the robot the capacity to perform novel actions, and is efficient in high dimensional spaces, and as a mechanism for communication [30]. It also speeds up the learning time by reducing the repetitions required for trial-and-error learning [31], and it can lead to open-ended learning without previous knowledge of the tasks or the environment [32].

Demonstration (also called self-imitation) [33, 34] avoids the problem of mapping from teacher to observer space. While this problem exists during imitation, it is eliminated in demonstration, as the human directly move the limbs of the robot [20] thus avoiding the "Correspondance Problem" [28]. It also does not require expert-knowledge of the domain dynamics, allowing the teacher to be a non- expert [20].

Some authors have also studied multi-modal learning, combining these techniques; including imitation and instructions [35-37] or demonstration and instruction [38]. In this research we build upon and extend these multi-modal approaches. We implement a multi-modal learning architecture which allow a user to teach action to robots (iCub and Nao) using one or a combination of language instructions, demonstration or imitation. More precisely, demonstration is a form of "tele-operation" by "kinesthetic teaching" and imitation is mediated by "external sensor" as defined in [20]: demonstration by kinesthetic teaching because the teacher operates directly on the robot learner platform, and imitation by external sensor because we are using kinect as perceptual device to encode the executing body's moves.

Thus the novelty of the current research is threefold – first it demonstrates a rich language capability for establishing and negotiating shared plans in real time. Second, it does this by allowing a multi-modal combination of spoken language programming, imitation and demonstration based learning. Finally, it demonstrates that, with an appropriate robotic platform, language can be used as the glue that binds together learning from these different modalities. These capabilities are demonstrated on two robots, the Nao and the iCub, which allow us to take advantage of the specific motor capabilities of each, including the more dexterous manipulation capabilities of the iCub.

II. System requirements and design

The goal of the current research is to demonstrate that a learning system that is based on the human developmental capability to map language onto action can provide the basis for a multimodal shared plan learning capability. In order to proceed with this analysis, we consider a scenario that involves multimodal learning. This will allow us in particular to determine the requirements involved in a human-robot cooperation to achieve an unknown task with real-time learning.

Consider a scenario where a humanoid robot and a human

are in a face-to-face interaction, with a box and a toy put on a table. The human wants to clean the table, by putting the toy in the box. In order to do that, he must first grasp the toy, then open the box, then put the toy in the box, and finally close the box. Let us further consider that the human cannot grasp the toy and open the box at the same time, and that he thus needs help in performing this task. The human will ask the robot to "clean the table". The robot doesn't yet know the plan so it will ask the human to explain. The user will describe each step of the plan, which is composed by several sequential actions:

- "I grasp the toy, then
- you open the box, then
- I put the toy in the box, then
- you close the box",

After checking whether the stated shared plan has been understood correctly, the robot will check each action that it should perform. The robot recognizes that there are some problems because it doesn't know how to open or close the box. It will ask for the help of the human, who has to teach it however he wants.

For opening the box, the human will decompose the teaching in two parts: at first, going to a safe initial position and next imitating him. After the opening action is learned, the user will teach the closing behavior, by directly demonstrating the motion by moving the arm of the robot. Finally, the robot has learned the whole shared plan and each action it should perform, and so the two agents can proceed and clean the table together. This scenario allows us to identify the functional requirements for the system. The system should:

1. Understand human language, including mapping grammatical structure onto internal representation of action.
2. Appropriately distinguish the definition of self and the other for relative pronouns (e.g. "I", "You"),
3. Manage a memory of known shared plan and actions,
4. Become active in the discussion by asking human when a problem occurred,
5. Perform Inverse kinematics mapping to learn from human action by imitation,
6. Encode proprioception induced when the human is moving the robot to teach.
7. Perceive the state of objects in the world.

In the following sections we will define an overall system architecture that accommodates requirements 1 – 4 in a platform independent manner, suggesting that these are the core learning functions. We will further demonstrate how this system can be used for real-time multimodal shared plan learning on the Nao with requirements 5 and 6, and on the iCub with point 7.

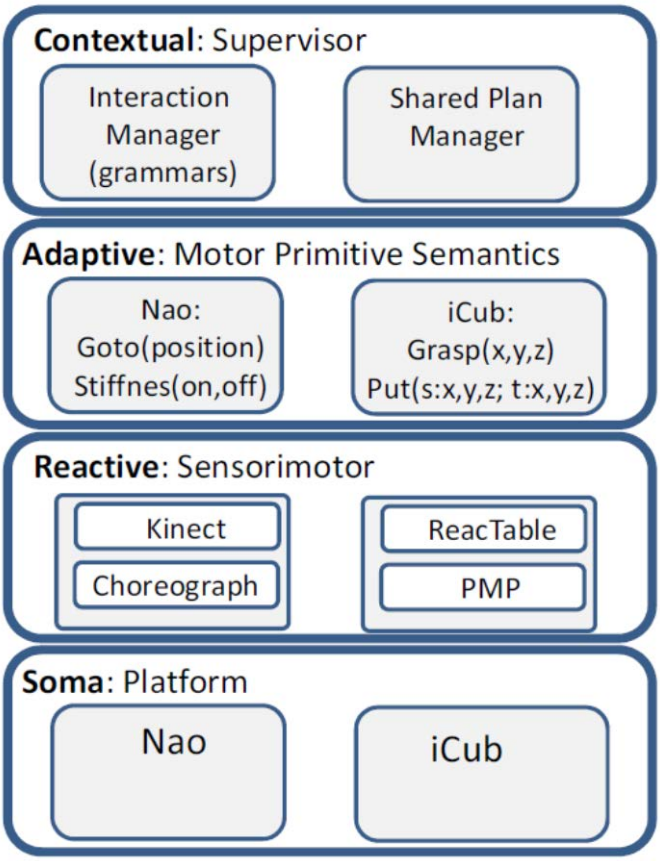


Figure 1. Biomimetic Architecture for Situated Social Intelligence Systems (BASSIS).

III. SYSTEM DESIGN OVERVIEW

Here we present the system architecture for the learning and execution of cooperative shared plans. We begin with the components that are independent of the physical platform, and then introduce the platform specific components.

The BASSIS architecture (Figure 1) is a multi-scale architecture organized at three different levels of control—reactive, adaptive and contextual, with the different levels of self are all based on the physical instantiation of the agent through its body (soma). It is based on the Distributed Adaptive Control Architecture [39-41]. Soma corresponds to the physical platform, instantiated as the Nao or iCub in our experiments. The Reactive or sensorimotor layer employs Kinect for perception and Choreograph™ (Aldebaran) for motor control on the Nao, and the ReacTable sensitive table, and the passive motion planner (PMP) and iKin inverse kinematic solver for iCub. The Adaptive layer defines adaptive motor capabilities for each robot. In the current context this adaptation can take place through learning within the human-robot interaction. The Contextual layer is platform independent, and implements a Supervisor function, with a grammar-based Interaction Manager, and a Shared Plan Manager. Within the BASSIS framework, the Contextual layer implements a form of long term memory that we exploit here in the context of learning shared action plans.

A. Supervisor

The Supervisor function consists in two related capabilities. The first is general management of the human-robot interaction via a state-based dialog management capability. The second is capability to learn and execute shared plans. Both of these functions are platform independent.

1) Interaction Management

Interaction Management is provided by the CSLU Toolkit [42] Rapid Application Development (RAD) state-based dialog system which combines state-of-the-art speech synthesis (*Festival*) and recognition (Sphinx-II recognizer) in a GUI programming environment. RAD allows scripting in the TCL language and permits easy and direct binding to the YARP domain, so that all access from the Interaction Management function with other modules in the architecture is via YARP.

The system is state based, with states for specifying the shared plan, modifying the shared plan, if there are errors, teaching specific actions within the shared plan, and finally, executing the shared plan during the cooperative task execution. Interaction management also allows the system to indicate error states to the user, and to allow him to explore alternate possibilities to rectify such errors, as illustrated in section IV.A.2.

2) Shared Plan Learning

The core aspect of of the learning capability is the capability to learn and execute shared plans, and to learn constituent actions that can make up those plans. As defined above, a shared plan is a sequence of actions with each action attributed to one of two agents in a turn-taking context. Shared plans can be learned via two complimentary learning mechanisms. The first method involves a form of spoken language programming, in which the user verbally describes the succession of action-agent components that make up the shared plan. Recognition is based on a grammar that we have developed for this purpose:

```
(1) $SharedPlan = pedro%% *sil%% $agent $command
    [<($linkWord $agent $command)>];
(2) $agent = you| I;
(3) $command =
        a. $action1 [*sil%%]  |
        b. $action2 [*sil%%]  ;
(4) $pause = [*sil%%] [*any%%] [*sil%%];
(5) $object = winnie | toy | chest;
(6) $posture = initial-position ;
(7) $action1 =
        a. grasp $pause $object|
        b. reach $pause  $object|
        c. open $pause  $object|
        d. close $pause  $object|
        e. move-to $pause $posture;
(8) $action2 = put $pause $object $pause [in%%] $pause
    $object;
(9) $linkWord = then |  after-that | next | and ;
```

Line (1) specifies that a shared plan begins with the "imperative" "Pedro" (the robot's name) followed by an optional silence (*sil%%), then an agent and command, followed by [0-n] groups made of a link word, an agent and a command. Agent terminals are identified in (2). Commands can take 1 or two arguments, as specified respectively in (7) and (8). Interestingly, in this grammar, the set of terminal nodes (actual words to be recognized) is only 16 distinct words. Thus, the speaker independent recognition system is in a well-defined recognition niche, and the system works with few to no errors.

In the case that errors are made, either in recognition, or by the user forgetting a command, saying a wrong command etc. we have a "spoken language programming" editing capability. Editing can involve the following edits: Replace one command with another. In this case the user repeats the faulty command, and then the correct one (in cooperation with the dialog system of the robot). Delete a command, in which case the user stats the command to be deleted. Insert a command, in which case the user says before or after a given command, and then the new command.

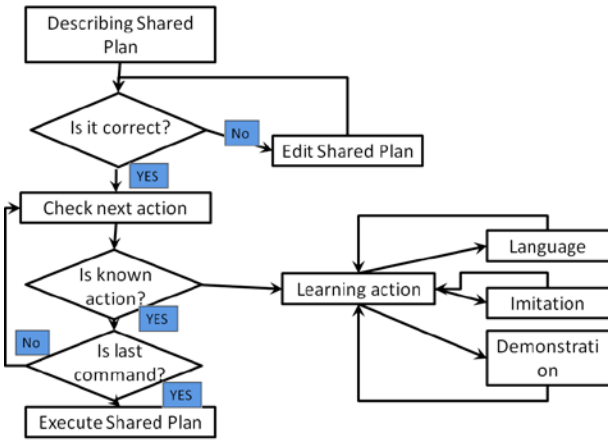


Figure 2. Shared Plan Manager. In the initial state, the user describes the entire shared plan. The robot repeats the understood plan, allowing editing. Then, for each action, if it is unknown, the system enters specific states for learning by language, imitation or demonstration. When all actions are learned, the shared plan is executed.

The second learning mechanism is evoked at the level of individual actions, and allows the user to teach new component actions to the robot. This involves a combination of spoken language programming and perceptual action recognition. Perceptual action recognition can occur via action recognition with the Kinect, and via kinesthetic demonstration, which will be detailed below. The robot can then use the resulting shared plan to take the role of either agent, thus demonstrating the crucial role-reversal capability that is the signature of shared planning [1, 12].

As illustrated in the example dialog with the Nao below, this provides a rich capability to negotiate a complex cooperative task using spoken language. The resulting system can learn how to perform novel component actions (e.g. open, close), and most importantly, it can learn arbitrary novel turn-taking sequences – shared plans – that allow the user to teach in any novel cooperative behavior to the robot in real-time. The only constraint is on the set of composite actions from which the novel behavior can be constructed.

B. YARP

Software modules in the architecture are interconnected using YARP [43], an open source library written to support software development in robotics. In brief YARP provides an intercommunication layer that allows processes running on different machines to exchange data. Data travels through named connection points called ports. Communication is platform and transport independent: processes are not aware of the details of the underlying operating system or protocol and can be relocated at will across the available machines on the network. More importantly, since connections are established at runtime it is easy to dynamically modify how data travels across processes, add new modules or remove existing ones. Interface between modules is specified in terms of YARP ports (i.e. port names) and the type of data these ports receive or send (respectively for input or output ports). This *modular* approach allows minimizing the dependency between algorithm and the underlying hardware/robot; different hardware devices become interchangeable as long as they export the same interface.

C. Humanoid Robot Nao and Kinect

The Nao (Figure 3) is a 25 degree of freedom humanoid robot built by the French company Aldebaran. It is a medium size (57 cm) entertainment robot that includes an onboard computer and networking capabilities at its core. Its open, programmable and evolving platform can handle multiple applications. The onboard processor can run the YARP server (described below) and can be accessed via telnet connection over the internet via WiFi.

More specifically, the Nao is equipped with the following : CPU x86 AMD Geode with 500 MHz, 256 MB SDRAM and 1 Gb Flash memories, WiFi (802.11g) and Ethernet, 2 x 640x480 camera with up to 30 frames per second, inertial measurement unit (2 gyro meters and 3 accelerometers), 2 bumper sensors and 2 ultrasonic distance sensors.

In this research, we extend the perceptual system of the Nao to include a 3D motion capture capability implemented with the Kinect ™ system. The Kinect recognizes a human body image in a configuration posture (see Fig. 3), and then continuously tracks the human body. Joint angles for three degrees of freedom in the shoulder and one in the elbow are extracted from the skeleton model, and mapped into the Nao joint space to allow real-time telecommand of the two arms.

D. iCub Humanoid and ReacTable Perceptual System

The iCub is a 53 DOF humanoid platform developed within the EU consortium RobotCub. The iCub [44] is an open-source robotic platform with morphology approximating that of a 3½ year-old child (about 104cm tall), with 53 degrees of freedom distributed on the head, arms, hands and legs. The current work was performed on the iCubLyon01 at the INSERM laboratory in Lyon, France. The head has 6 degrees of freedom (roll, pan and tilt in the neck, tilt and independent pan in the eyes). Three degrees of freedom are allocated to the waist, and 6 to each leg (three, one and two respectively for the hip, knee and ankle). The arms have 7 degrees of freedom, three in the shoulder, one in the elbow and three in the wrist. The iCub has been specifically designed to study manipulation, for this reason the number of degrees of freedom of the hands has been maximized with respect to the constraint of the small size. The hands of the iCub have five fingers and 19 joints.

1) Motor Control

Motor control is provided by PMP. The Passive Motion Paradigm (PMP) [45] is based on the idea of employing virtual force fields in order to perform reaching tasks while avoiding obstacles, taking inspiration from theories conceived by Khatib during 80's [46]. Within the PMP framework it is possible to describe objects of the perceived world either as obstacles or as targets, and to consequently generate proper repulsive or attractive force fields, respectively. A meaningful example of attractive force field that can be produced is the so called spring-mass-damper field; in this case the relevant parameters are the stiffness constant and the damping factor, which regulate the force exerted by a target placed in a given spatial location. An effective model that represents repulsive force fields is the Multivariate Gaussian function, which accounts for a field centred at an obstacle and is characterized by the typical bell-shaped decay. According to the composition of all active fields, the manipulator's end-effector is eventually driven towards the selected target while bypassing the identified obstacles; evidently, its behaviour and performances strictly depend on the mutual relationship among the tuneable field's parameters.

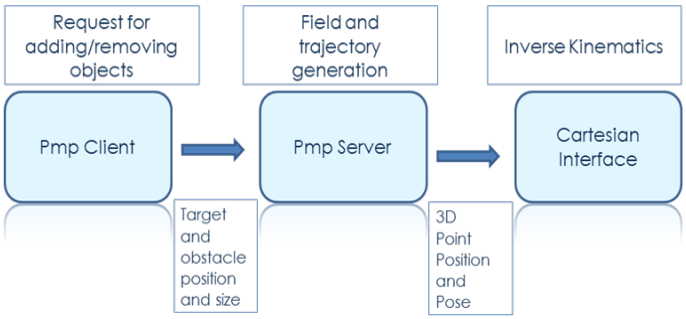


Figure 3: PMP software architecture.

However, in order to tackle the inverse kinematics problem and compute the final trajectory of the end-effector, the original PMP makes use of the Transposed Jacobian algorithm; this method is well known to suffer from a number of weaknesses [47] such as the difficulty to treat constraints of complex kinematic structures as the iCub arm turns to be [48, 49]. Therefore, we have decided to replace the Transposed Jacobian approach with a tool that relies on a powerful and fast nonlinear optimizer, namely Ipopt [50]; the latter manages to solve the inverse problem while dealing with constraints that can be effectively expressed both in the robot's configuration space (e.g. joints limits) and in its task-space. This new tool [49] represents the backbone of the Cartesian Interface, the software component that allows controlling the

iCub directly in the operational space, preventing the robot from getting stuck in kinematic singularities and providing trajectories that are much smoother than the profiles yielded by the first implementation of PMP.

In this changed context, the Cartesian Interface lies at the lowest level of the revised PMP architecture, whose simplified diagram is show in Figure 3. At higher level the pmpServer element is responsible of composing the final force field according to the objects currently stored in an internal database. Users can add, remove or modify this database in the easiest way by forwarding requests to the server through a dedicated software interface, made available by the pmpClient component. It is important to point out that the properties of objects stored in the database can be retrieved for modification in real-time in order to mirror the environment as it evolves over time. All the software components of the revised PMP architecture can be openly accessed from the iCub repository.

2) Perception

In the current research we extend the perceptual capabilities of the iCub with the ReacTable™. The ReacTable is licensed by Reactable Systems. The ReacTable has a translucid surface, with an infrared illumination beneath the table, and detection system that perceives tagged objects on the table surface with an accuracy of ~5mm. Thus, tagged objects can be placed on the table, and their location accurately captured by the infrared camera.

Interaction with the external world requires that the robot is capable of identifying its spatial reference frame with the objects that it interacts with. In the human being, aspects of this functionality is carried out by the dorsal stream, involving areas in the posterior parietal cortex which subserve complex aspects of spatial perception [51]. In our system, the 2D surface of the table is calibrated into the joint space of the iCub by a linear transformation calculated based on a sampling of three calibration points on the table surface that are pointed to by the iCub. Thus, three points are physically identified in the Cartesian space of the iCub, and on the surface of the ReacTable, thus providing the basis for calculation of a transformation matrix which allows the projection of object coordinates in the space of the table into the Cartesian space of the iCub. These coordinates can then be used as spatial arguments to the PMP action system of the iCub, described above, which provides basic physical actions including point-to(x, y, z), put(source X, Y, Z; target x, y, z), grasp(x, y, z), release(x, y, z).

IV. Multimodal Learning I: Nao & Kinect

Here we demonstrate the system functionality for real-time learning, in which language, spatial perception (via Kinect), and proprioception are combined to allow the Nao to learn a shared plan which is then executed in real-time with the human teacher. The idea is to solve a common problem encountered during daily life which is cleaning a table with some objects on it. As defined in the requirements analysis in section II, the goal is to place the object from the table to a chest. However, the chest needs to be opened before doing that, and we need to close it afterward to complete the task.

A. Learning the Shared Plan and Component Actions

Here we present the time-tagged unfolding of the cooperative dialog where the shared plan is established, and indicate how the system components are engaged to permit the corresponding interaction, and learning and execution of the shared plan.

1) Establishing the shared plan

The user begins by giving the name of the cooperative shared plan:

`00:25 Nao: Tell me what I have to do.`
`00:26 Human: “Pedro, help me to clean the table”.`

The Nao recognizes the spoken order and checks in the shared plan manager if it knows this plan, and determines that there is no definition for such a plan...

`00:30 Nao: “The shared plan is clean.`
`00:36 Nao: “Please explain me.”`

The shared plan manager creates an empty shared plan called *“clean the table”* and it is ready to be filled with the human description.

The user verbally explains all the steps of the shared plan, including both moves of the robot and himself. It is worth noting that the shared plan is initially specified in ~10 seconds.

`00:37 Human: “I reach the toy, then I reach the trash box then you open the trash box then I put the toy in the trash box then you close the trash box”.`

The Nao will segments the global shared plan in a sequential set of 3 actions using the grammar built in the spoken language interface, and then repeats it to be sure it has perfectly understood the whole interaction :

`00:50: Nao: “I reach the toy, you reach the trash box, then I open the trash box, you put the toy in the trash box, I close the trash box. Do you agree with this plan?”`

Two remarks should be made. First, note that in order to be intuitive and understandable, for relative pronouns “You” or “I” rather than repeating these literally, the Nao has created the plan by replacing the relative agent by the absolute ones: the human for “I” and the Nao for “You” because the human was speaking.

2) Correcting the Shared Plan

The second remark here is that there is in fact an error in the plan. Nao states that it should reach for the toy, rather than the human doing so. When the Nao is repeating the plan, it is not only to check if it has understood the orders but it also provides the human a way to verify his plan.

The user has here the possibility to edit the shared plan (section IIA2), and will do it by answering that he does not agree with this, for instance:

`Human: “No I do not”.`

Thus the robot will ask where the problem is:

01:05 Nao: ”So, What did I misunderstand?“.

Here, the human has several ways to edit the plan:

- He could specify that there is a problem with a specific action by repeating it and could, after that, remove it or replace it by another one,
- He could erase everything and describe the plan again from scratch,
- He could insert a missing action.

In this dialogue example described below, the problem is with the agent of the first action. The Nao said “I reach the toy”, but the intended action was “You reach the toy”, with “you” referring to the human, so in response to the robot’s question “what did I misunderstand?”, the human names the erroneous action:

01:08 Human: “I reach the toy”

Thus, this dialog is motivated by the user’s recognition that an error has occurred, followed by his specification of the erroneous component. This will trigger an editing of the plan, with the Nao asking

01:10 Nao: “What do you want me to do instead?”

A new command can now be inserted to replace the erroneous one.

01:13 Human: “I reach the toy”.

Figure 4. Multimodal learning. (Left column) Using perceptual based imitation, the human demonstrates to the Nao how to open the box, and the Nao performs the action in real-time, mirroring the action of the human. (Right column) Using kinesthetic demonstration the human teaches the Nao how to close the box.

This illustrates the detailed manipulation of relative pronouns, as the human is now speaking from his own perspective, and thus refers to himself as “I”. To confirm the edit, the Nao then places a “virtual cursor” after the element to be replaced, and before the next action, and verifies that the new command should be inserted there.

01:15 Nao: “So where – I reached the toy, instead, you will reach the toy. Do you agree with this plan?

01:24 Human: “Yes, I do”.

The robot now has a complete shared plan including tasks for the user and for itself. The next step is to verify that all of the tasks assigned to the robot are known.

3) Learning component actions

The Nao iterates through the shared plan until it encounters an action assigned to it, determines if it knows the action, and informs the user if not. A query is made to the Motor Primitive Library where the list of current known behaviors of the Nao is stored into. There may be several unknown behaviors in a new plan, thus to keep the interaction more animated the system selects one of the three interrogatory sentences at random.

Here, the Nao recognizes that it has to perform the “open” action, which is unknown for now and will call the human for help:

01:26 Nao: “Behavior ‘open’ is unknown. I am going to learn ‘open’”

This will put the supervisor in the learning state. The human could create the behavior in the Motor Sequence Learning with one or a combination of these methods (see Figure 2):

- Calling a known behavior by language,
- Asking the robot to imitate him,
- Physically demonstrating the action by moving the robot’s arms

As illustrated in Figure 4 (left column), the chest can be opened by pushing a lever, and the human will choose to teach “open” by decomposing the action in two steps: First, the robot needs to move its arms up in the air, before, second, going down to push the lever. The “put-the-hands-up” is a known action, so the human will command this action.

01:34 Nao: “Tell me what to do.”

01:36 Human:“Move to Kinect init”.

This is recognized to be a known behavior in Motor Primitive Library, so the robot asks for a confirmation before performing the action:

01:38 Nao: “I am going to move to Kinect init. Are you sure?”.

01:42 Human: “Yes I am”.

After the user’s acknowledgment, Nao executes the behavior and the resulting joint angle trajectory will be stored in the new “open” behavior of Motor Sequence Learning to begin the construction of this action.

The robot informs the user that the action is finished and it is waiting for the following command.

01:51 Nao: “Tell me what to do”.

The human proceeds to the second step of the teaching by calling the imitation method:

01:53 Human:“Imitate me”.
01:56 Nao: “Imitation activated. Tell me when it is finished”.

The system then enters the state (Figure 2) where joint angles are read from the Kinect device based on the observed human demonstrator (which is calibrated in the hands up position). The user then demonstrates the motion required to open the box, using visual feedback from the Nao in real-time. The corresponding joint angle trajectory is concatenated to those from the previous method. The user then terminates the imitation.

02:17 Human: “It is finished”.

The “open” behavior has thus been learned by the Nao in real-time as a complex joint trajectory, and is transferred from Motor Sequence Learning to Motor Primitive Library so that it is now available as a command.

Following the flow outlined in Figure 2, the Nao proceeds to the next behavior it should perform in the shared plan, and again there is an unknown action: “close”.

02:20 Nao: “The behavior ‘close’ is unknown. I am going to learn ‘close’”.
02:27 Nao: “Tell me what to do.”

An empty “close” behavior is created in Motor Sequence Learning. This time, the human chooses to teach the action using the demonstration method.

02:29 Human: “Let me move your arms”.

This places the supervisor in a state where it commands the Nao to set the stiffness of the arms to zero so that they can be manipulated by the human.

02:33 Nao: “Stiffness changed. Tell me when it is finished”.

As illustrated in Fig. 3 (right column) the human then physically demonstrates the action, pushing the Nao’s hand and arm through a trajectory that closes the box. Once the demonstration is finished, the human terminates the learning, and asks the robot to proceed with the shared plan execution.

02:46 Human : “It is finished”.

Thus, this dialog fragment, and Figure 4, illustrate two complimentary learning methods: imitation of an action trajectory performed by the human, and physical demonstration where the human physically guides the learned trajectory of the robot. Depending on the nature of what is to be learned, the user can choose the more appropriate method.

B. *Executing the Shared Plan*

The shared plan is now fully specified, and the Nao begins execution.

02:56 Nao: “The current step is you reach toy. You are supposed to reach. Tell me when it is done”.
03:01 Human: “It is done”.
03:03 Nao: “The current step is I open trash box”.

Here the robot uses the learned action trajectory to reproduce the “open” movement.

03:27 Nao: “The current step is you put the toy trash box. You are supposed to put. Tell me when it is done”.
03:34 Human: “It is done”.
03:37 Nao: “The current step is I close trash box”.

The Nao then uses the joint trajectory that was taught by kinesthetic demonstration, to close the box. At this point the toy has successfully been closed in the box, and the shared plan is finished.

03:52 Nao: “This was the last command”.

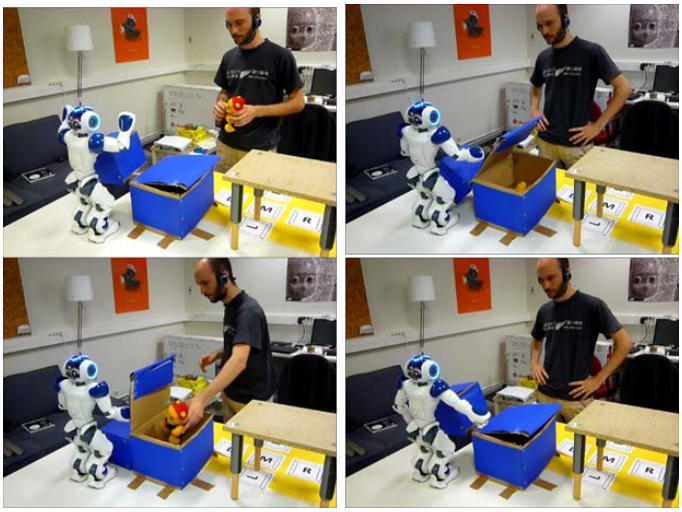


Figure 4. Shared plan execution. Left column: Human takes toy, Nao opens box, human places toy in box. Right column: Nao closes box.

C. *Performance Analysis*

We analyze performance from three separate executions of the learning task described above. Two were performed in the laboratory, and the third was performed during the Robocup @home Open Challenge 2011 in Istanbul, July 2011. In this case, we were required to install and set up the system in 3 minutes, and then had five minutes to perform the task, with no possibility to shift to a different time, or to have another 5 minutes in case of failure. The task was successfully completed, and our “Radical Dudes” team placed 4th/19 in the Open Challenge. This demonstrates the robustness of the system.

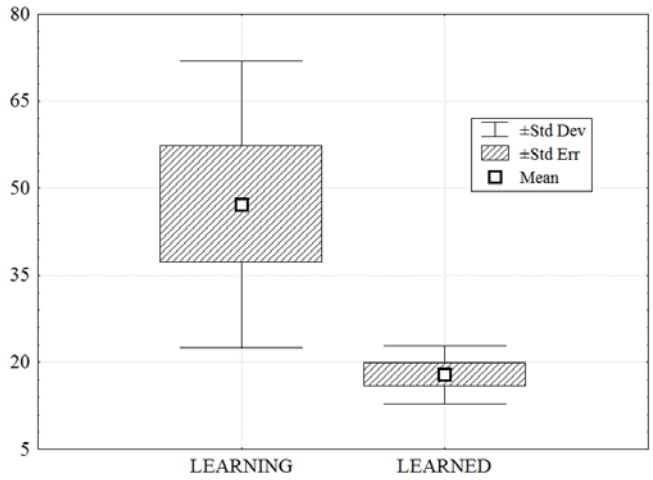


Figure 5. Effects of shared plan learning on overall action execution time in seconds.

For each of the three sessions where the shared plan was learned and then executed, we measured the time to complete the open-the-box and close-the-box actions during the learning phase, and then during execution of the learned shared plan. Execution time is measured from the onset of the human command, to the execution of the action and onset of next request by the Nao. Thus, during learning, the execution time includes the teaching component. In order to compare the effect of learning on the time to complete individual actions, we performed non-parametric Wilcoxon signed-rank test comparing each action when it was being learned vs. when it had been learned, collapsing across sessions. There were two actions per session (open and close), each performed once in learning and once in execution after learning. With the three sessions, this provided a total of 6 learning-learned comparisons. As illustrated in Figure 5, there is a significant reduction in execution time during the shared plan execution. This was confirmed in a significant learning effect in the Wilcoxon signed-rank test, N = 6, $Z = 2.20$, $p = 0.027$. We thus demonstrated that the system can learn to produce arbitrary sequences of actions with a turn-taking structure. The principle limiting factor is simply the set of basic level actions from which the shared plans can be constructed. Three repetitions of the "clean-up" shared plan, including one during the Robocup@Home Open Challenge, demonstrate the reliability of the system. Over these three trials, we also demonstrated a significant effect of this learning (as opposed to simply commanding the robot) in terms of behavior execution time after learning.

D. *Nao Experiment Discussion*

We have previously demonstrated how the user can employ language to teach new actions [13, 14], and then combined the previously learned actions into a new shared plan [11, 15]. The current research extends this shared plan learning. For the first time, we demonstrate how spoken language can be used to coordinate on-line multimodal learning for a shared cooperative plan. The multiple modalities include imitation of actions performed by the human (using the Kinect), human demonstration of a desired trajectory by physically manipulating the robot arm, and finally, spoken language based invocation of known actions, with all of these modalities contributing to a coherent and integrated plan. We should stress that learning by imitation, demonstration and spoken language programming all have been extensively studied in the past. What is new here is the combination of these multiple modalities in a coherent and seamless manner, under the control of language. It is worth noting that while we emphasize the learning of the "clean the table" shared plan, the grammar-based learning capability allows for the construction of arbitrary turn-taking action sequences.

V. MULTIMODAL LEARNING II: ICUB & REACTABLE

While the learning that we observed in the previous section has certain components that are platform specific (e.g. the morphology of the Nao, and the mapping of the Kinect to that morphology), the principal learning component which is based on spoken language is platform independent. Indeed, it provides a method for the linking together of action primitives into shared plans that can subsequently be used to achieve cooperative activity.

A. *Learning the Shared Plan and Component Actions*

In the first experiment with the iCub, the human explains a shared cooperative task similar to that used with the Nao, where the goal is to "hide" a toy under a box. The grammar is of the same structure as that used for the Nao. The principal difference is that it is constructed so that the plan is successively constructed from single actions that are concatenated with the previous actions (4).

(1) $subjects = I | Me | You | Stephane | Maxime | Peter | iCub;
(2) $objects = box | toy | trumpet | drums | left | middle | right;
(3) $action =
  a) $subjects [*sil%% | *any%%] grasp [*sil%% | *any%%] $objects |
  b) $subjects [*sil%% | *any%%] point [*sil%% | *any%%] $objects |
  c) $subjects [*sil%% | *any%%] put [*sil%% | *any%%] $objects [*sil%% | *any%%] $objects |
  d) $subjects [*sil%% | *any%%] uncover [*sil%% | *any%%] $objects [*sil%% | *any%%] $objects;
(4) $sharedPlan = concatenate($sharedPlan,$action) ;

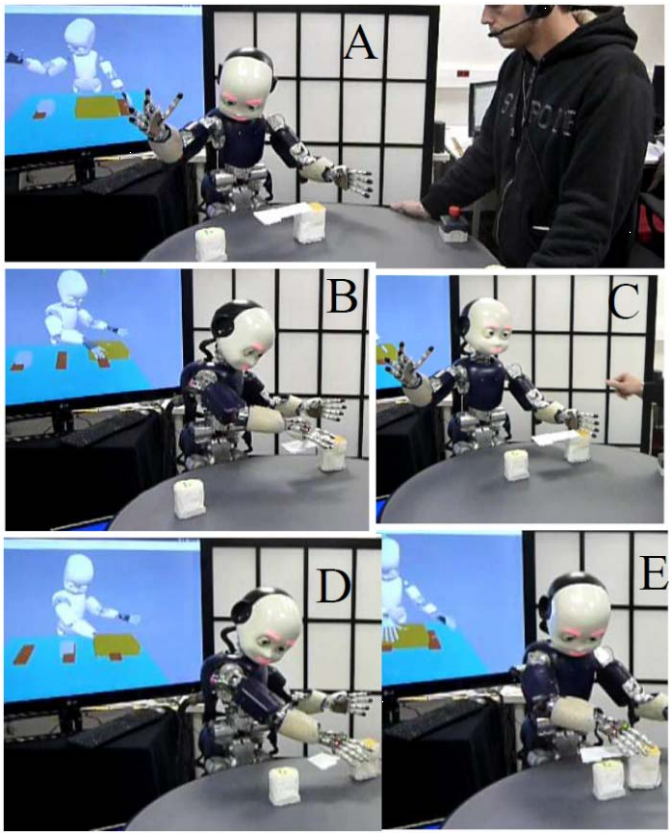


Figure 6. Learning and performing the "hide the toy" cooperative plan. Setup with the "toy" on the left of the table, and "box" on the right. Spatial representation on iCub GUI left rear. B. iCub puts the box on its left. C. human put the toy in the middle. D. iCub reaches for the box, and F. puts the box on the toy. Note the grasping precision.

Here we illustrate how by exploiting this grammar, the shared plan manager can construct the shared plan from the sequence of actions described by the subject.

00:00 ICub: "What can I help you with?"
00:02 Human: "Shared Plan".
00:05 ICub: "Shared Plan menu".
00:07 Human: "You and I hide the toy with the box".

The Supervisor consults the Shared Plan Manager, and determines that there is no definition for this plan.

00:11 ICub: "Tell me how you and I can hide toy box?"
00:13 ICub: "What is next?"
00:17 Human: "You put box left".
00:20 ICub: "Did you say 'you put box left'?".
00:23 Human: "Yes I did".
00:25 ICub: "What is next?".
00:27 Human:"I put toy middle".
00:30 ICub: "Did you say 'I put toy middle'?".
00:32 Human: "Yes I did".
00:35 ICub: "What is next?".
00:38 Human:"You put box toy".
00:40 ICub: "Did you say 'You put box toy'?".
00:43 Human: "Yes I did".
00:46 ICub: "What is next?".
00:48 Human: "Finished".

The successive actions of the shared plan have now been extracted, and the plan is stored as:

Hide (you, I, toy, box){
(You put (box left))
(I put (toy middle))
(You put (box toy))}

In the definition, the four arguments to hide are instantiated with the "you, I, toy, box". When the shared plan is used in the future, it can be called with different arguments, e.g. hide(John, you, trumpet, carton) for "John and you hide the trumpet with the carton", and these new arguments will be systematically substituted into the component actions of the shared plan. This provides the ability generalize to new objects and agents, including providing a capability for role reversal as specified in section C.

B. Executing the Shared Plan

01:00 Human: "You and I hide the trumpet with the box"

Note here that the system allows arguments to be used in the "hide" shared plan. Here trumpet will be hidden.

01:04 ICub: "I will put the box on the left. You will put the trumpet on the middle. I will put the box on the trumpet. Should we proceed?".
01:24 Human: "Yes we should"
01:26 ICub: "I will put the box on the left".

The robot performs its action, illustrated in Figure 6A-B.

02:02 ICub: "You will put the trumpet on the middle".

The human makes the move, and the robot detects the change of position of the trumpet to the middle (Figure 6C).

02:12 ICub: "I will put the box on the trumpet".

The iCub performs the final move, covering the trumpet with the box (Figure 6D-E), completing the shared plan..

02:48 ICub: "Shared Plan menu".

C. Role Reversal

In the cooperative task, the user can decide who takes which role. This implies the principal of role reversal. As stated in the introduction, role reversal is taken as evidence that the agent has a global representation of the actions that both agents will perform. This representation allows the agent to take on the role of either of the two participants in the shared plan [12].

In order to reverse roles, the human can use the same shared plan, and indicate role reversal by specifying the arguments in the reversed order. He can say "you and I hide the toy in the box" or "I and you hide the toy in the box." As specified in section V.A above, depending on how the pronouns I and you are ordered in the specification, the roles of the human and robot will be reversed.

```
00:08 Human: "I and You will hide the
   toy in the box"
00:13 ICub: "You will put the box on
   the left. I will put the toy on the
   middle. You will put the box on the
   toy. Should we proceed?".
00:29 Human: "Yes we should"
00:31 ICub: "You will put the box on
   the left".
```

Here the robot detects the change of position of the box to the left.

```
00:37 ICub: "I will put the toy on the
   middle".
01:13 ICub: "You will put the box on
   the toy".
01:19 ICub:  "Shared Plan menu".
```

Role reversal is a specific instance of a more general capability that is provided by the system. That is, once a shared plan has been learned with a given set of agent arguments, the arguments for the two agents can be instantiated with different instances, e.g. I and you vs. you and I.

D. Performance Analysis

We repeated the shared plan learning, execution and role reversal twice each. The timing of the principal events is illustrated in Figure 7. It is noteworthy that the system allows the multiple-action shared plan to be specified in well-under one minute, and then executed in this same time scale, both in the standard format, and the role reversal.

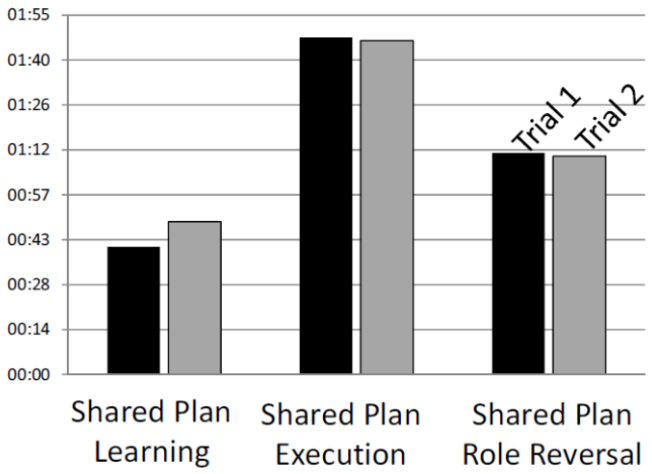


Figure 7. Event durations (in seconds:minutes) for two trials (Trial 1 in black, trial 2 in grey) of the learning, execution and role reversal for the "hide the toy" shared plan.

Note that in Figure 7, the role reversal condition is executed more rapidly than the standard condition. This is due to the relative slowness of the robot actions, with respect to those of the human. In the standard sequence, the robot performs two actions (moving the box away from the center, and then over the toy) while the human performs only one action (placing the toy in the middle to be covered). This is reversed in the role, reversal, and thus the effect of the slowness of the robot is reduced.

E. iCub Discussion

These experiments extend the results with the Nao, which is in part achieved because of the more dexterous grasping capabilities of the iCub. In the current experiments we demonstrated how an arbitrary shared plan could be established in less than one minute, and then immediately be used to execute the cooperative task. In addition, we demonstrate how this shared plan can be used to allow role reversal, in which the two agents swap roles. Again, for Carpenter et al. [12] this is a hallmark of shared plan use, as it clearly demonstrates that the agents have a "bird's eye view" or global view, of the shared activity. Technically this requires that all of the actions that can take place in the shared plan can be executed physically by both the human and the robot. Because of the high spatial precision of the ReacTable, and the precision grasping capabilities of the iCub, this is a technical reality.

VI. DISCUSSION AND FUTURE WORK

The current research can be situated within the larger context of cognitive developmental robotics [52], with physical embodiment playing a central role in structuring representations within the system, through interaction with the environment, including humans. In development, the early grammatical constructions that are acquired and used by infants define structural mappings between the underlying structure of everyday actions, and the expression of this structure in language [6, 53]. We have exploited this

mapping, in building systems that can learn grammatical constructions from experience with the environment [8, 54]. Here we exploit this type of grammatical construction, by building such constructions into the grammars that are used for speech recognition. These constructions that map onto the basic structure of action (e.g. *agent action object*) correspond to the basic argument constructions that are the workhorses of initial language [6, 53]. The "ditransitive" construction is a good example that has been extensively studied [5]. In a canonical form of this construction "`Subject Verb Recipient Object`" (e.g. John gave Sally a flower), `Subject` maps onto the agent of the transitive action specified by `Verb`, and `Recipient` receives the `Object` via that transitive action. The current research demonstrates how language, based on these constructions, can be used to coordinate real-time learning of cooperative actions, providing the coordination of multiple demonstration modalities including vision-like perception, kinesthetic demonstration [13, 29, 55-58], and command execution via spoken language. In this sense, language serves a dual purpose: First and most important, it provides the mechanism by which a cooperative plan can be constructed and modified. Second, during the construction of the shared plan, one of the modalities by which actions can be inserted into the plan is via the spoken issue of a command. We demonstrate that in this framework, the constructive features of language can be mapped onto different robot platforms. This requires the mapping of the argument structure of grammatical constructions onto the predicate-argument structure of the command and perceptual operators of the given platform [13, 55]. Doing so, we subsequently achieve performance, where the systems can learn and perform new cooperative behaviors in the time frame of 2-3 minutes. The introduction of structured language provides a powerful means to leverage sensory-motor skills into cooperative plans, reflecting how the development of language in human children is coincident with an explosion in their social development in the context of triadic relations between themselves, another person and a shared goal [1]. We should note that the "ecological validity" of the kind of language that the user can employ is somewhat restricted to simple grammatical constructions. That is, people cannot use fully unconstrained natural language, such as relative clauses, and pronouns. Still, this allows sufficient expressive ability for the user to construct elaborated shared plans.

The approach to learning that we have taken thus consists in the implementation of a highly structured scaffolding that allows the user to teach the robot new action components, and then to teach the robot how to organize these actions into more elaborate turn-taking sequences that constitute shared plans. The advantage of this approach is that it is powerful and scales well. It is powerful because it allows the user to specify arbitrary turn-taking sequences (which can even include solo sequences that are performed only by one of the agents), and the set of elementary actions can also be augmented through learning. All of this learning can be done with a single trial. The advantage of this is that learning is rapid. Indeed, related studies have demonstrated that for complex tasks such as those used here, human and neural network simulations fare better with high level instruction (imitation or verbal instruction) than with lower level instruction (reinforcement learning) [59]. The disadvantage is that the teaching must be perfect. Thus, in demonstrating a trajectory, the system cannot benefit from a successive refinement over multiple trials [60].

One of the limitations of this work is that there is not a systematic mechanism for the long-term accumulation and synthesis of such learning. In the future it will be important for these developmental acquisitions to be integrated into the system over a life-time scale [61]. Another limitation is that in the current research the behavior is determined by the shared plan, and there is no choice. To cope with changing task contingencies, the system will require more adaptive behavior including the ability to choose between competing options [62]. Perhaps one of the most fundamental limitations of the current research, which lays a foundation for future research, has to do with the deeper nature of the shared plan. This is the notion of the shared intention. Our robots can learn a plan that allows them to perform a cooperative task, and event to demonstrate role reversal. Yet the true notion of the actual final goal, the shared intention, to get that toy into the box, is currently not present. We have started to address this issue by linking actions to their resulting states, within the action representation [56]. We must go further, in order to now expand the language capability to address the expression and modification of internal representations of the intentional states of others.

The current research proposes an interaction architecture, for on-line multi-modal learning, and demonstrates its functionality. It is not an extended user study that allows for the collection of data whose variability can be statistically analyzed in a population of subjects. Within the interactions that we test, the most pertinent parameter that reflects the change in the real-time flow and fluidity of the interactions is related to the time required for different component actions, and their changes as a function of learning. We thus demonstrate the feasibility of using spoken language to coordinate the creation of arbitrary novel turn-taking action sequences (which we refer to as shared plans). This includes the ability to create new actions (through demonstration and imitation), and to embed these actions in new turn-taking shared plans. Clearly a more robust demonstration of the performance of the architecture (and effective time gains before/after learning) should use naïve users and include metrics related to interaction quality, success etc. This is a topic of our ongoing research.

ACKNOWLEDGMENT

This research has been funded by the European Commission under grants EFAA (ICT-270490) and CHRIS (ICT-215805).

REFERENCES

[1] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll, "Understanding and sharing intentions: The origins of cultural

cognition," *Behavioral and Brain Sciences*, vol. 28, pp. 675-691, 2005.
[2] S. Carey, *The Origin of Concepts*. Boston: MIT, 2009.
[3] F. Warneken, F. Chen, and M. Tomasello, "Cooperative activities in young children and chimpanzees," *Child Development*, vol. 77, pp. 640-663, 2006.
[4] F. Warneken and M. Tomasello, "Helping and cooperation at 14 months of age," *Infancy*, vol. 11, pp. 271-294, 2007.
[5] A. Goldberg, *Constructions: A Construction Grammar Approach to Argument Structure*. Chicago: University of Chicago Press, 1995.
[6] M. Tomasello, *Constructing a language: A usage based approach to language acquisition*. Boston: MIT Press, 2003.
[7] K. Rohlfing and J. Tani, "Grounding Language in Action," *IEEE Transactions on Autonomous Mental Development*, vol. 3, p. 4, 2011.
[8] P. Dominey and J. Boucher, "Learning to talk about events from narrated video in a construction grammar framework," *Artificial Intelligence*, vol. 167, pp. 31-61, 2005.
[9] P. Dominey, A. Mallet, and E. Yoshida, "Progress in programming the hrp-2 humanoid using spoken language," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2169-2174.
[10] P. Dominey, A. Mallet, and E. Yoshida, "Real-time cooperative behavior acquisition by a humanoid apprentice," in *International Conference on Humanoid Robotics*, Pittsburg, Pennsylvania, 2007.
[11] P. Dominey and F. Warneken, "The basis of shared intentions in human and robot cognition," *New Ideas in Psychology*, vol. 29 p. 14, 2011.
[12] M. Carpenter, M. Tomasello, and T. Striano, "Role reversal imitation and language in typically developing infants and children with autism," *Infancy*, vol. 8, pp. 253-278, 2005.
[13] S. Lallée, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, T. van Der Tanz, F. Warneken, and P. Dominey, "Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception," in *IROS*, Taipei, 2010.
[14] S. Lallée, U. Pattacini, J. Boucher, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, K. Hamann, J. Steinwender, E. A. Sisbot, G. Metta, R. Alami, M. Warnier, J. Guitton, F. Warneken, and P. F. Dominey, "Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions," in *IROS*, San Francisco, 2011, pp. 2895 - 2902.
[15] S. Lallée, U. Pattacini, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, K. Hamann, J. Steinwender, E. A. Sisbot, G. Metta, J. Guitton, R. Alami, M. Warnier, T. Pipe, F. Warneken, and P. Dominey, "Towards a Platform-Independent Cooperative Human-Robot Interaction System: III. An Architecture for Learning and Executing Actions and Shared Plans," *IEEE Transactions on Autonomous Mental Development*, vol. In press, 2012.
[16] H. H. Clark, "Coordinating with each other in a material world," *Discourse Studies*, vol. 7, pp. 507-525, October 1, 2005 2005.
[17] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469-483, 2009.
[18] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," in *Proceedings of the International Conference on Robotics and Automation*, 1996, pp. 2700-2705.
[19] M. Nicolescu and M. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, 2003, pp. 241-248.
[20] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469-483, 2009.
[21] P. Dominey, A. Mallet, and E. Yoshida, "Real-Time spoken-language programming for cooperative interaction with a humanoid apprentice," *Intl J. Humanoids Robotics*, vol. 6, pp. 147-171, 2009.
[22] V. Tikhanoff, A. Cangelosi, and G. Metta, "Integration of Speech and Action in Humanoid Robots: iCub Simulation Experiments," *Autonomous Mental Development, IEEE Transactions on*, vol. 3, pp. 17-29, 2011.
[23] Y. Zhang and J. Weng, "Task Transfer by a Developmental Robot," *Evolutionary Computation, IEEE Transactions on*, vol. 11, pp. 226-248, 2007.
[24] P. Andry, P. Gaussier, S. Moga, J. P. Banquet, and J. Nadel, "Learning and communication via imitation: an autonomous robot perspective," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 31, pp. 431-442, 2001.
[25] P. Andry, P. Gaussier, and J. Nadel, "From Visuo-Motor Development to Low-level Imitation," 2002.
[26] C. A. Calderon and H. Hu, "Robot Imitation from Human Body Movements," in *In Proceeding AISB05 Third International Symposium on Imitation in Animals and Artifacts*, 2005.
[27] S. Calinon, F. D'Halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and Reproduction of Gestures by Imitation," *Robotics & Automation Magazine, IEEE*, vol. 17, pp. 44-54, 2010.
[28] K. Dautenhahn and C. L. Nehaniv, "The Correspondence Problem," ed: MIT Press, 2002.
[29] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics and Autonomous Systems*, vol. 54, pp. 361-369, 2006.
[30] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, pp. 481-487, 2002.
[31] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Sciences*, vol. 3, pp. 233-242, 1999.
[32] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, pp. 151-190, 2003/12/01 2003.
[33] J. Saunders, C. L. Nehaniv, and K. Dautenhahn, "Using Self-Imitation to Direct Learning," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, 2006, pp. 244-250.
[34] J. Saunders, C. L. Nehaniv, K. Dautenhahn, and A. Alissandrakis, "Self-imitation and Environmental Scaffolding for Robot Teaching," *International Journal of Advanced Robotic Systems*, vol. 4, 2008.
[35] A. Cangelosi and T. Riga, "An Embodied Model for Sensorimotor Grounding and Grounding Transfer: Experiments With Epigenetic Robots," *Cognitive Science*, vol. 30, pp. 673-689, 2006.
[36] K. Coventry, A. Cangelosi, R. Rajapakse, A. Bacon, S. Newstead, D. Joyce, and L. Richards, "Spatial Prepositions and Vague Quantifiers: Implementing the Functional Geometric Framework
Spatial Cognition IV. Reasoning, Action, Interaction." vol. 3343, C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, Eds., ed: Springer Berlin / Heidelberg, 2005, pp. 98-110.
[37] J. J. Steil, F. Röthling, R. Haschke, and H. Ritter, "Situated robot learning for multi-modal instruction and imitation of grasping," *Robotics and Autonomous Systems*, vol. 47, pp. 129-141, 2004.
[38] J. Weng, "Developmenta Robotics: Theory and Experiments," *International Journal of Humanoid Robotics (IJHR)*, vol. 1, pp. 199-236, 2004.
[39] P. F. Verschure, T. Voegtlin, and R. J. Douglas, "Environmentally mediated synergy between perception and behaviour in mobile robots," *Nature*, vol. 425, pp. 620-4, Oct 9 2003.
[40] A. Duff, M. S. Fibla, and P. F. Verschure, "A biologically based model for the integration of sensory-motor contingencies in rules and plans: a prefrontal cortex based extension of the Distributed Adaptive Control architecture," *Brain Res Bull*, vol. 85, pp. 289-304, Jun 30 2011.
[41] P. F. Verschure and T. Voegtlin, "A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: Distributed Adaptive Control III," *Neural Netw*, vol. 11, pp. 1531-1549, Oct 1998.
[42] S. Sutton, R. Cole, J. Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, and K. Shobaki, "Universal speech tools: The CSLU toolkit," in *Fifth International Conference on Spoken Language Processing*, 1998.
[43] P. Fitzpatrick, G. Metta, and L. Natale, "Towards Long-Lived Robot Genes," *Robotics and Autonomous Systems*, vol. 56, pp. 29-45, 2007.
[44] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *PerMIS: Performance Metrics for Intelligent Systems Workshop*, Washington DC, USA, 2008, pp. 19-21.
[45] V. Mohan, P. Morasso, G. Metta, and G. Sandini, "A biomimetic, force-field based computational model for motion planning and

bimanual coordination in humanoid robots," *Autonomous Robots*, vol. 27, pp. 291-307, 2009.
[46] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, pp. 90-98, 1986.
[47] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, 2005.
[48] A. Parmiggiani, M. Randazzo, L. Natale, G. Metta, and G. Sandini, "Joint Torque Sensing for the Upper-Body of the iCub Humanoid Robots," in *IEEE International Conference on Humanoid Robots*, Paris, France, 2009.
[49] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An Experimental Evaluation of a Novel Minimum-Jerk Cartesian Controller for Humanoid Robots," in *IROS*, Taipei, 2010.
[50] A. Wächter and L. T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, pp. 25-57, 2006.
[51] L. Shmuelof and E. Zohary, "Dissociation between ventral and dorsal fMRI activation during object and action recognition," *Neuron*, vol. 47, pp. 457-470, 2005.
[52] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive Developmental Robotics: A Survey," *IEEE Trans Autonomous Mental Development*, vol. 1, p. 22, 2009.
[53] A. Goldberg, *Constructions: A construction grammar approach to argument structure*. Chicago: University of Chicago Press, 1995.
[54] P. Dominey and J. Boucher, "Developmental stages of perception and language acquisition in a perceptually grounded robot," *Cognitive Systems Research*, vol. 6, pp. 243-259, 2005.
[55] S. Lallée, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, T. van Der Tanz, F. Warneken, and P. Dominey, "Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions," presented at the IROS, 2011.
[56] S. Lallée, C. Madden, M. Hoen, and P. Dominey, "Linking language with embodied teleological representations of action for humanoid cognition," *Frontiers in Neurobotics*, 2010.
[57] S. Lallée, F. Warneken, and P. Dominey, "Learning to collaborate by observation," in *Epirob*, Venice, 2009.
[58] G. Metta, G. Sandini, L. Natale, L. Craighero, and L. Fadiga, "Understanding mirror neurons: a bio-robotic approach," *Interaction studies*, vol. 7, pp. 197-232, 2006.
[59] F. Dandurand and T. R. Shultz, "Connectionist Models of Reinforcement, Imitation, and Instruction in Learning to Solve Complex Problems," *IEEE Trans Autonomous Mental Development*, vol. 1, p. 11, 2009.
[60] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans Syst Man Cybern B Cybern*, vol. 37, pp. 286-98, Apr 2007.
[61] Y. Demiris and A. Meltzoff, "The Robot in the Crib: A Developmental Analysis of Imitation Skills in Infants and Robots," *Infant Child Dev*, vol. 17, pp. 43-53, Jan 2008.
[62] T. J. Prescott, F. M. Montes Gonzalez, K. Gurney, M. D. Humphries, and P. Redgrave, "A robot model of the basal ganglia: behavior and intrinsic processing," *Neural Netw*, vol. 19, pp. 31-61, Jan 2006.

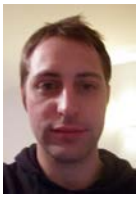
Maxime Petit received the M.Sc. degree in computer sciences (cognitive sciences specialty) from the University of Paris-Sud (Orsay), France, in 2010 and an engineering degree in biosciences (bio-informatics and modeling specialty) from the National Institute of Applied Sciences (INSA) of Lyon, France, the same year.

He is currently a PhD student at the Stem-Cell and Brain Research Institute, from the unit 846 of the National Institute of Science And Medical Research (INSERM) in Bron, France, working in the Robotic Cognition Laboratory team. He is focusing about the reasoning and planning in robotics, especially with the iCub platform, and more precisely, within a context of spoken language interaction with a human.

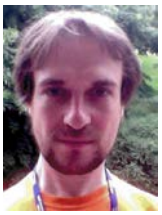
Stéphane Lallée received the master's degree in Cognitive Science and Human-Machine Interface Engineering from the University of Grenoble in 2008. He joined the Robot Cognition Laboratory in 2008, and has played a leading role in the iCub project in Lyon since the arrival of the iCub in 2009. He received the PhD in Cognitive Neuroscience from Lyon University in 2012, developing a distributed architecture for human-robot cooperation.

Jean-David Boucher received the PhD in Cognitive Science and Robotics from theUniversity of Lyon, in 2010, at the Robot Cognition Laboratory, INSERM, Lyon France, and then continued there as a post-doctoral fellow on the CHRIS project. He currently teaches programming at Monash University in Melbourne Australia, and pursues research on human-robot interaction.

Grégoire Pointeau received the M.Sc. degree in biosciences (bio-informatics and modeling specialty) from the National Institute of Applied Sciences (INSA) of Lyon, France, in 2011, and then pursued Mater's studies in Cognitive Science at the University of Lyon. He is currently a Ph.D. student at the Robot Cognition Laboratory in Lyon as part of the FP7 EFAA project. He is interested in the accumulation of experience and autobiographical memory in the elaboration of the self, in the context of human-robot social interaction.



Pierrick Cheminade received a Technical Diploma in Computer Sciences from the University Institute of Technology, Puy-en-Velay, France in 2008. Before working in the game industry he was an independent web developer. He began his career in 2008 at DreamonStudio (Lyon, France), and then worked for 2 years at Little World Studio (2008-2011 Lyon France) before working at the Robot Cognition Laboratory, on the Nao, at the Inserm-U846 (2011). In 2012 he released several games as an independent on the Ios platform. He is currently working as a game programmer for Codemaster (Southam United Kingdom).



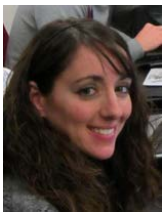
Dimitri Ognibene graduated in Computer Engineering from the University of Palermo (Italy, 2004) and received his Ph.D. in Robotics from the University of Genoa (Italy, 2009). He was Research Assistant the CNR ISTC in Rome from 2005 until 2011. In 2010 he was visiting scholar at the University of Massachussets Amherst. He is currently a Research Associate at Imperial College London. His research interests are the computational principles of visual attention, resource allocation, and learning and development in natural and artificial systems. He is now working on probabilistic social attention systems for humanoid robots.

Eris Chinellato (M'03-11) received his Ph.D. in Intelligent Robotics from Jaume I University (Spain, 2008), his MSc in Artificial Intelligence, with the Best Student Prize, from University of Edinburgh (UK, 2002), and his Industrial Engineering Degree from Universit`a degli Studi di Padova (Italy, 1999).

His interdisciplinary research, integrating robotics with experimental and theoretical neuroscience, focuses on sensorimotor integration in both natural and artificial systems.

He is now working at Imperial College London on neural models of social interaction to be applied to the iCub humanoid robot. Dr. Chinellato has published in influential journals and proceedings in both robotics and neuroscience, and has served as reviewer and program committee member for several IEEE journals and conferences.

Ugo Pattacini is a postdoctoral fellow in the Robotics, Brain and Cognitive Sciences Department (RBCS) at the Istituto Italiano di Tecnologia (IIT) in Genova, Italy. He holds an M.S. with honors (in 2001) in Electronic Engineering from University of Pisa and a Ph.D. (in 2011) in Robotics, Neurosciences and Nanotechnologies from IIT. From 2001 to 2006 he worked as embedded software developer for Formula 1 applications first at Racing Department of Magneti Marelli in Milan and then joining Toyota F1 team in Cologne, dealing with the design and the implementation of proprietary hard-constrained real-time operating systems, vehicle dynamics and torque-based traction control strategies. From 2006 to 2007 he moved to Earth Observation Business Unit of Thales Alenia Space in Rome where he was concerned with the specifications design and trade-off analyses of the satellite data acquisition system for COSMO-SkyMed (ASI) and GMES (ESA) scientific program. From 2008 he is involved in the development of the humanoid iCub at IIT focusing his interests mainly on the advancement of robot motor capabilities and pursuing a methodology that aims to combine traditional model-based approaches with the most recent machine learning techniques.



Ilaria Gori was born in Rome, Italy on November, 6th 1986. She completed a Bachelor Degree in Management Engineering with the score of 110/110 at Sapienza, University of Rome, in July 2008. In 2010 she graduated (cum laude) in Computer Engineering at Sapienza, University of Rome, with a specialization in Artificial Intelligence.
She wrote her master thesis at Imperial College, in London, where she lived for 6 months in 2010. She published a paper at ISVC 2010 summarising her master thesis, and she won a Best Paper Award. Then she worked for two months in a software development company in 2010. In January 2011 she started a PhD in Robotics, Cognition and Interation Technologies at Istituto Italiano di Tecnologia, Genova, Italy, that she is currently carrying out. She is mainly interested in computer vision and machine learning, but she also worked in robotics.

Uriel Martinez-Hernandez is a PhD student in Automatic Control and Systems Engineering department at the University of Sheffield. His current research is about exploration and object recognition based on active touch sensing using the fingers of the ICUB robot. Also in this project he is working together with Psychology department.

Hector Barron is an Engineer with a first degree in Computer Science, I worked several years in applying computer vision and machine learning upon real-time systems. Attracted by cognitive science, I am proposing increasing the capacities of spatial reasoning for navigation tasks. Upon the Bayesian framework, this work explores interesting insights from neuroscience and psychology.

Martin Inderbitzin received the PhD in Computer Science and Robotics in the Synthetic, Perceptive, Emotive and Cognitive Systems (SPECS) Group, 2007 – 2012 in Barcelona. He studied Embodied Models of Emotions - Verification of Psychological and Neurobiological Theories of Emotions Using Virtual and Situated Agents. He is currently R&D Director Projektil Pot Shot Experience Design, Zürich, Switzerland, where he will investigate video mapping and interactivity developing embodied experiences

Andre Luvizotto Ph.D. student at the SPECS group at the Universitat Pompeau Fabra, Barcelona, Spain. He joined SPECS in February, 2008. He studied in Brazil where he received a bachelor in music (2005) and a MsC degree on Electrical Engineering (2008), both in UNICAMP, with the cooperation of NICS - " Núcleo Interdisciplinar de Comunicação Sonora". In his master thesis, he worked with sonological models, based on wavelet transform. He is currently working on the EFAA project, with a computational model for visual and auditory invariant representations, based on temporal population code.

Vicky Vouloutsi (Athens - Greece, 1982) studied Computer Science at the Technological Educational Institute of Athens (BA 2008). In 2009 she came to Barcelona where she did her Master in Cognitive Systems and Interactive Media at Universitat Pompeu Fabra (MSc 2011). She completed her Master Thesis on biologically inspired computation for chemical sensing (www.neurochem-project.org) in SPECS where she is currently working as a continuation of her Master Thesis.

Yiannis Demiris (M'01-SM'08) received the B.Sc (1994) and Ph.D (1999) degrees in intelligent robotics from the University of Edinburgh, Edinburgh, U.K. He joined the faculty of the Department of Electrical and Electronic Engineering at Imperial College London in 2001 where he is currently a Reader in human–centred robotics and heads the Personal Robotics Laboratory. His research interests include biologically inspired robotics, human–robot interaction, developmental robotics, and robotic assistive devices for adults and children with disabilities..
Dr. Demiris was the Chair of the IEEE International Conference on Development and Learning in 2007 and the Program Chair of the ACM/IEEE International Conference on Human–Robot Interaction in 2008. He has organized several international workshops on robot learning, bio-inspired machine learning, and epigenetic robotics. He has received fellowships from the AIST–MITI in Japan, and the European Science Foundation, and currently participates in several EU FP7 research projects in Human-Robot Interaction. In 2012 he received the Rector's Award and the Faculty of Engineering Award for Excellence in Engineering Education.

Giorgio Metta is senior scientist at the IIT and assistant professor at the University of Genoa where he teaches courses on anthropomorphic robotics and intelligent systems for the bioengineering curricula. He holds a MS with honors (in 1994) and PhD (in 2000) in electronic engineering both from the University of Genoa. From 2001 to 2002 he was postdoctoral associate at the MIT AI-Lab where he worked on various humanoid robotic platforms. He is assistant professor at the University of Genoa since 2005 and with IIT since 2006. Giorgio Metta's research activities are in the fields of biologically motivated and humanoid robotics and in particular in developing life-long developing artificial systems that show some of the abilities of natural systems. His research developed in collaboration with leading European and international scientists from different disciplines includnig neuroscience, psychology, and robotics. Giorgio Metta is author or co-author of approximately 100 publications. He has been working as research scientist and co-PI in several international and national funded projects. He has been reviewer for international journals, national and international funding agencies.

Peter Ford Dominey is a CNRS Research Director at the INSERM Stem Cell and Brain Research Institute in Lyon France, where he directs the Robot Cognition Laboratory. He completed the BA at Cornell University in 1984 in cognitive psychology and artificial intelligence. In 1989 and 1993 respectively he obtained the M.Sc. and Ph.D. in computer science from the University of Southern California, developing neural network models of sensorimotor sequence learning, including the first simulations of the role of dopamine in sensorimotor associative learning, and pioneering work in reservoir computing. From 1984 to 1986 he was a Software Engineer at the Data General Corporation, and from 1986 to 1993 he was a Systems Engineer at NASA/JPL/CalTech. In 1997 he became a tenured researcher, and in 2005 a Research Director with the CNRS in Lyon France. His research interests include the development of a "cognitive systems engineering" approach to understanding and simulating the neurophysiology of cognitive sequence processing, action and language, and their application to robot cognition and language processing, and human-robot cooperation. He is currently participating in several French and European projects in this context.