Towards a distributed, embodied and computational theory of cooperative interaction

Stéphane Lallée Lallée

N° d'ordre : 52-2012 Année 2012

THESE DE L'UNIVERSITE DE LYON

Délivrée par

L'UNIVERSITE CLAUDE BERNARD LYON 1

ECOLE DOCTORALE NEUROSCIENCES ET COGNITION

DIPLOME DE DOCTORAT
(arrêté du 7 août 2006)

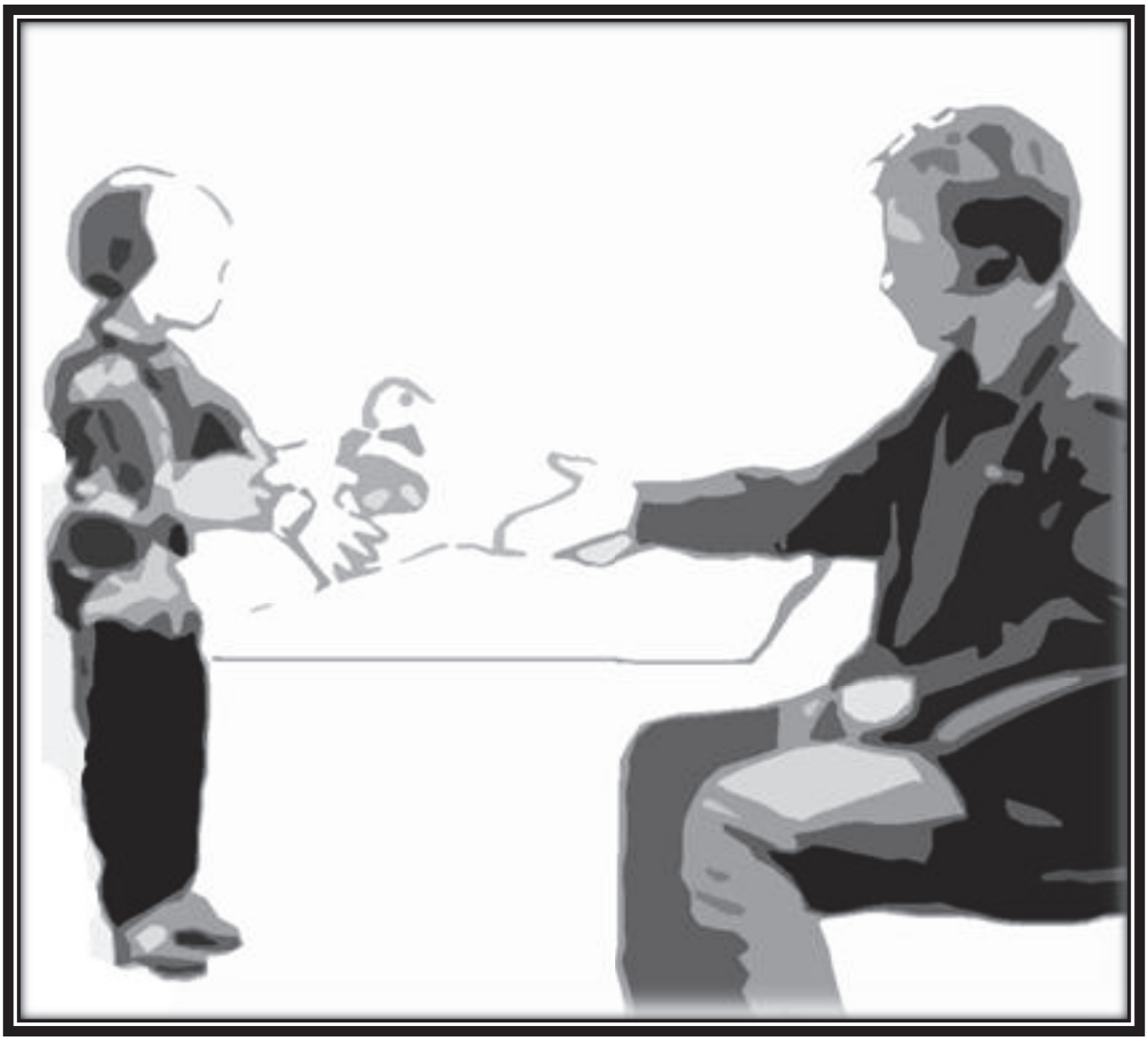
Soutenue publiquement le 02 Avril 2012 par

LALLEE Stéphane

TOWARDS A DISTRIBUTED, EMBODIED AND COMPUTATIONAL THEORY OF COOPERATIVE INTERACTION.

Pr Remi Gervais	Président du jury
Dr Giorgio Metta	Rapporteur
Dr Paul Vershure	Rapporteur
Dr Henry Kennedy	Membre du jury
Dr Peter Ford Dominey	Directeur de thèse

TOWARDS A DISTRIBUTED, EMBODIED AND COMPUTATIONAL THEORY OF COOPERATIVE INTERACTION



I would like to thank all the people that contributed to this work, can it be by helping me with fundings, with teaching me knowledge, pushing me in the right direction or just supporting me in an emotional way. All of this couldn't have been achieved without you all…

RESUME

Les robots vont peu à peu intégrer nos foyers sous la forme d'assistants et de compagnons, humanoïdes ou non. Afin de remplir leur rôle efficacement ils devront s'adapter à l'utilisateur, notamment en apprenant de celui-ci le savoir ou les capacités qui leur font défaut. Dans ce but, leur manière d'interagir doit être naturelle et évoquer les mêmes mécanismes coopératifs que ceux présent chez l'homme. Au centre de ces mécanisme se trouve le concept d'action : qu'est-ce qu'une action, comment les humains les reconnaissent, comment les produire ou les décrire ? La modélisation de toutes ces fonctionnalités constituera la fondation de cette thèse et permettra la mise en place de mécanismes coopératifs de plus haut niveau, en particulier les plan partagés qui permettent à plusieurs individus d'œuvrer de concert afin d'atteindre un but commun. Finalement, je présenterai une différence fondamentale entre la représentation de la connaissance chez l'homme et chez la machine, toujours dans le cadre de l'interaction coopérative : la dissociation possible entre le corps d'un robot et sa cognition, ce qui n'est pas imaginable chez l'homme. Cette dissociation m'amènera notamment à explorer le « shared experience framework », une situation dans laquelle une cognition artificielle centrale gère l'expérience partagée de multiples individus ayant chacun une identité propre. Cela m'amènera finalement à questionner les différentes philosophies de l'esprit du point de vue de l'attribution d'un esprit à une machine et de ce que cela impliquerai quant à l'esprit humain.

ABSTRACT

Robots will gradually integrate our homes wielding the role of companions, humanoids or not. In order to cope with this status they will have to adapt to the user, especially by learning knowledge or skills from him that they may lack. In this context, their interaction should be natural and evoke the same cooperative mechanisms that humans use. At the core of those mechanisms is the concept of action: what is an action, how do humans recognize them, how they produce or describe them? The modeling of aspects of these functionalities will be the basis of this thesis and will allow the implementation of higher level cooperative mechanisms. One of these is the ability to handle "shared plans" which allow two (or more) individuals to cooperate in order to reach a goal shared by all. Throughout the thesis I will attempt to make links between the human development of these capabilities, their neurophysiology, and their robotic implementation. As a result of this work, I will present a fundamental difference between the representation of knowledge in humans and machines, still in the framework of cooperative interaction: the possible dissociation of a robot body and its cognition, which is not easily imaginable for humans. This dissociation will lead me to explore the "shared experience framework, a situation where a central artificial cognition manages the shared knowledge of multiple beings, each of them owning some kind of individuality. In the end this phenomenon will interrogate the various philosophies of mind by asking the question of the attribution of a mind to a machine and the consequences of such a possibility regarding the human mind.

Table of content

Preface

Robots. Before addressing any specific, focused and specialized sub-topics, this thesis is about robots. From books, cinema and video games, robots are already present in our imagination, somehow known by people. However, robots living an imaginary world of science fiction are not the ones that I know. I have to start this thesis by a cynical statement: robotics is an illusion.

Doing a PhD in robotics can be seen mostly as spending countless hours of writing austere code behind a computer screen, but it is also conducting a constant social and psychological experiment each time someone questions you about your job. What I discovered through this dialog is that working as a "cognitive roboticist" is similar to being an illusionist. When one goes to a magic show, he really sees that the assistant girl is cut in two pieces and glued again after a while. His eyes send the information to his brain about what is going on in front of him, and a part of his brain believes that the girl is cut. However, if you ask him after the show if he actually thinks that the lady was sliced in half, he will say "No! Of course there is a trick!". In the case of robots, I observed an even more paradoxical situation. When I have to do a demonstration to naïve people, or journalists, I'm always astonished by how easy it is to "trick" them.

A simple dialog between you and the machine in English and one will ask you "*Does he understand French as well?*". I'd like to answer "*Yes, he can understand and speak all languages in the world*", however the only thing that I can honestly say is *"It doesn't understand English. It's only translating a sequence of sounds to a text in English, this text triggers a text reply in English which is translated back into a sound that you truly understand*". The fact is that the Chinese Room experiment (Searle 1980) is less popular than Wall-E...

Another funny question often raised is "*If you had to compare it to a child, how old would your robot be?*". Well, this kind of child is able to process differential equations in a matter of nano seconds, while at the same time destroying his whole arm by pushing it

through a wooden table that it doesn't perceive and without feeling any pain. It is a child that can record his whole life in high definition video, while not being able to tell you anything about what he did in the past day. What age would you give to such a child?

Examples I could produce to make my point are legion: *"He is looking at you! He likes you!", "He looks so sad!", "Which toy does he prefer?"…* The fact is that people naturally assign agency, feelings and human-like intelligence to a (humanoid) robot. Every single motion or sound made by the robot is instantly imbued with intentionality, suggesting that the observer unconsciously assigns some high level cognition to the machine. This is what happens during the show, people believe in "magic" because their own brains suggest to them much more than they actually see. But then, after the show, comes this question that I used to ask to naïve subjects: "*Do you think that a robot can be intelligent?*" or even a more delicate one "*Do you think that a robot can be conscious*?". The answer is always a too strong "*No*", served with plenty of justifications involving often a fuzzy concept of "soul" and a raging indignation that we can even think about machines being our equal. My point is not to evoke a precocious remake of the Valladolid debate[1] but to highlight a few points about the upcoming integration of robots into our society. First is that the illusion of intelligence is born easily in the eye of humans; while at the same time they are firmly convinced that this very thing they experience cannot be true.

The main difference between difference between the robotics researcher and the illusionist stands in their approach of building the illusion. The magician uses a mind trick to present something that he knows to be a "lie" as a true fact. The robotic researcher tries to create real magic, not to have the people think he is flying, but to fly for real. When I design the brain of a robot, I do not want people to think it is intelligent, I want it to be so. This thesis is about my efforts helping to reach this goal, to be able to meet one day a robot that I will consider as being the equal of humans in terms of intelligence and consciousness. Agreeing on the fact that we are still at the babblings of robotics, my very hope is to see this day. It is both a hope and a fear. I fear that when this time comes, only the magician will know that his magic is real. Because for most of people, intelligence and consciousness are

[1] The Valladolid debate (1550-1551) is famous historical debate which purpose was to determine if America's natives had a soul or not and therefore if they could be subject to slavery or not.

something magic, and we can only recreate an illusion of magic. A hope, because mankind used to consider natural phenomena as magic, until they were explained by science and mastered by technology. So perhaps building a conscious machine will make people understand their own consciousness without considering it as a thing of magic, sacred or at least impossible to engineer.

Introduction

One starts to write this thesis using a highly technological device while drinking a homebrewed coffee. None of this could have been possible without the collaboration of hundreds of people. Even this activity, writing a thesis, is somehow part of a general cooperative plan: taking a step towards a higher level of knowledge in the EU Cognitive Robotics community. It is the result of a thinking process, which leads to ideas. Those ideas are written down in order to save them, to share them more easily with others, to avoid someone else wasting time in the thinking process. At a more practical level, we see results of cooperation everywhere: the machine which made the coffee, the water that arrived from the tube and the coffee beans that were planted, picked and roasted; all of those required at some point many people to work together.

Mankind has been able to achieve great things that are not within the reach of a single being (Tomasello 1999). Through cooperation, we are able to jump over a wall that is too high or to build the Great Wall. However, in both cases the same principles apply: a long chain of cognitive processes. At the "lowest" level, parts of our cortex refine our raw sensorial experience of the world. All our different sensory signals are merged through a learning of the correlations between elements composing the reality. In Chapter 1, I will describe a cortical model which allows a generic transformation from raw sensory data to the symbolic level. I will show that an object, an action or any concept can be represented as a pattern of cortical activity. The model described will be using a modified version of Kohonen Self Organizing Maps (Kohonen 1990), however it will be the only part of this thesis which will be about neural modeling. Having reached this point, I would like to speak about my position regarding neural networks. In the past, the artificial intelligence field has seen quite an opposition between classic AI (symbolic) and connectionism with (Fodor 1975) Before starting my PhD I was on the connectionist side, I believed that all problems in AI could be solved using neural networks and that it would be easier to model cognition this way. Having been working three years with a robot I still think that every problem can be solved using a neural model, however I also know that the optimal way to solve problems is not always the one that the nature choose. Neural networks are mathematical tools; they are good at

performing transformations between spaces when we do not have a clear idea about how to formalize this operation. It is the case when we try to model the perception: no model nowadays can provide us a full "symbolization" of the world experienced through the sensory apparatus of a robot, the symbol grounding problem is not yet solved(Harnad 1990). Therefore the best systems dealing in that field are mostly neuro inspired, even if their performances are too unpredictable. However when it comes to higher level cognitive functions, the brain processes are in some manner manipulating symbols, concepts. I'm not oversimplifying the cortical computations; I'm simply arguing that symbols are present in the brain, wielding the form of complex and distributed pattern of activity[2]. If we assume a complete symbolized representation of the real world then software engineering provides us many excellent ways to model cognition. Who never dreamt of being able to use a debugger to inspect the brain in a given state, having variables containing the "apple" or "eat" concepts instead of multidimensional vectors representing the same meaning? After the first chapter, I will assume that my models can be built among this symbolization and I will use mainly software engineering to model cooperation. This thesis will follow a bottom up plan, going from the lowest level of cognition (close to the sensors) to a fairly high level (cooperative interaction and language manipulation).

Chapter 2 will describe the concept of action. While it is quite easy to get an intuition about what the symbol of an object could be, the nature of an action is a bit more difficult to handle. Actions are not only a motor sequence, neither a simple sequence of behaviors. An action can be perceived, executed and described; it is deeply linked to the real world which makes it possible or not and which will be modified along its execution. An action is a way to change the world; it is the very first element which makes a body appearing to be animated. A stone could be the smartest thing in our universe while we would have no idea because it is unable to act. Since action is the main "building block" of any intelligent behavior, I will examine which insights are given by both neurophysiology and developmental psychology. Based on this literature, I present a data structure which can be used to generically represent an action in order to allow its recognition, execution and verbal description.

[2] The extensive problem of symbol grounding, and the academic debate concerning the challenges between symbolic and connectionist approaches will not be addressed in any detail here.

Moreover, I will present imitation, one of the basic requirements of cooperating abilities which is intrinsically matched when using this action model.

Once given a way to manipulate actions as symbols, it becomes possible to handle them in order to build a meaningful interaction between the robot and a human counterpart. In Chapter 3, I will focus on cooperation and again base my implementation work on studies performed in the fields of neurophysiology and developmental psychology. Along their development, children learn to interact with others in order to reach goals that are beyond their individual abilities. They demonstrate early an ability to share their goal with somebody else, using gaze, gestures and later on language. Once the goal is defined, a shared plan is constructed with the help of the partner; by assigning specific actions to each participant children find a path to reach their goal together. As mentioned above, having an exhaustive data structure (or representation) for action allows for their straight forward manipulation. Therefore a shared plan can be seen as a sequence of actions, each action being assigned to a specific agent. However, while this implementation is fairly simple, it is powerful: the goal of a shared plan can be determined by summing up all the goals of the component actions, the description and verbal negotiation for establishing the plan can be built using the actions specific descriptions, etc. Moreover, such a shared plan can be seen as a path between two world states. By knowing the current state of the world, and the desired goal, many planning algorithms can be used to create a shared plan. While this is not the point focused upon in this thesis, I will briefly discuss how to generate a goal directed shared plan using this information.

A prevalent approach to building cognitive systems today is to examine human brain function, and to attempt to mimic that process. It is a fair approach, since we have no clear alternative guides about how to build intelligent systems. However we have to keep in mind that human cognition is built on the top of the animal body limitations. In the last chapter of this thesis I will see “beyond the body” and describe which unique features an artificial cognitive machine can realize. Whether artificial symbols represent objects, actions or plans, they can be represented in a well defined software data structure which can be stored in memory, written in a file or exchanged through the network. The knowledge and cognitive working material of a robot is a collection of those representations, most of which is fully unembodied, opening up to manipulation of knowledge as if it was any other kind of data.

Chapter 4 will introduce the basis of an abstract (unembodied) cognitive machine with distributed bodies: while possibilities are endless, already a few achievements and problems are within our reach. I'll present them along with Central Cognition, a system designed to handle a centralized abstract cognitive machine while controlling multiple robotic bodies in parallel. Based on the possibility of such an architecture, I'll pursue a reflection about what could become a "Shared Experience Framework", applying and extending the Cartesian dualism to artificial cognitive machines and asking new questions about what are the mind, the body and the individuals. While this chapter will be mainly about the technical feasibility of such a framework, I'll pursue a deeper philosophical interrogation about those concepts within the final discussion of the thesis.

Chapter I

Embodied action, merging multiple sensory modalities

Introduction

The common approach to study the cortex is to divide it into areas (Brodmann 1909; Amunts, Schleicher et al. 1999) and to determine how those areas are connected both in term of anatomical (Felleman and Van Essen 1991; Markov, Ercsey-Ravasz et al. 2010) and functional connectivity (Cordes, Haughton et al. 2000). Although the neuroscience community doesn't always agree on the exact segmentation details, many researchers try to draw the connectivity map of the cortex (Braitenberg and Schüz 1998; Guye, Parker et al. 2003). From outside the community, one can imagine that the motivation guiding this research is that from the structure of the cortical network one could infer the functional dynamics and the respective role of each area. Due to historical reasons, the primary areas (vision mainly) which are "close to the sensors" have been studied more intensively than the rest of the cortex. They are generally thought as being organized in a hierarchical way (Felleman and Van Essen 1991), with leaves (the bottom levels of the hierarchy) being the areas closest to the sensors. Although cortex is clearly not a mathematical hierarchy (Markov, Ercsey-Ravasz et al. 2010; Vezoli, Gariel et al. 2010) there is a hierarchical flavor in its global organization: areas close to sensors merge into amodal zones which often send feedback to the bottom and continue to merge together upper in the stream[3]. This framework has built up the idea of convergence zones (Damasio 1989; Damasio and Damasio 1994). In a nutshell this theory holds that some cortical areas could act as pool of pointers to other areas, therefore linking several cortical network together. These zones would be responsible for linking together representations from various sensory modalities of the same concepts. A concrete example is that seeing a photo of a very dirty and wet dog could give you a sensation of its smell. The olfactive representation of such the odor associated to the dog in the picture could be activated because those two modalities (olfactive and visual) are linked in some high level conceptual convergence zone. This example is quite naïve and convergences zones are dealing which much more distributed and functional linking of concepts and functions; however the main idea is there: they merge networks of lower level cortical areas into amodal higher level concepts and solve this way the binding problem by allowing the extraction of units and regularities from the complex and not segmented raw sensor

[3] This is currently written in a naive manner. We will benefit from constructive input from the committee to improve these paragraphs.

information. The feedback process is also important; it could be a basis for explanations of certain perceptual illusions: we never perceive reality as the pure raw signal coming from our sensors; instead we perceive sensory information mixed with feedback data coming from higher levels in the stream. Indeed we do not perceive the world as it is, but as we think it is. Most illusions are based on the regularities our brain is used to experience in the world and on consistency between our different sensors. A well known evidence of this is the McGurk effect (McGurk and MacDonald 1976) which makes us "hear what we see", thus providing evidence that regularity in visuo-audio patterns shapes our perception. The convergence zone framework and the feedback influence could explain this effect in an elegant way as well as many other illusions. As it is quite generic and relatively easy to implement, the convergence zones framework served as a basis mainly for theoretical models (Moll and Miikkulainen 1997; Howe and Miikkulainen 2000). Moll's model provides a very good starting point for an implementation on a robot, but it lacks a major feature of cortical computation: topographical organization. It is a well-known phenomenon that some areas of the cortex will get activation in similar locations while presented two stimuli that are similar. Mostly studied within the visual cortex (Kosslyn, Thompson et al. 1995; Schall, Morel et al. 1995; Engel, Glover et al. 1997), this topographical organization also occurs in the motor cortex with the somatotopic (Buccino, Binkofski et al. 2001) mapping and the famous "homunculus" (Metman, Bellevich et al. 1993; Aflalo and Graziano 2006). Indeed it is quite appealing to consider that that this mechanism of convergence is quite generic and spread throughout the cortex, although no clear evidence of this has been systematically investigated. From a pure modeling point of view, this property is also interesting : topographical organization or neural maps allow an easy representation and understanding of what is going on in the network, which is one of the reasons that made the Self Organizing Map of Kohonen so famous (Kohonen 1990). In this chapter I will present a neural network model called Multi Modal Convergence Maps (MMCM) which fuses ideas from Kohonen's SOM and from the Convergence Zone Framework. It allows the learning and recall of multi-modal traces together with a spatially topographic storage in a self-organizing map of neurons. The model is used to process low level sensory information coming from the robot sensory apparatus and merge it into amodal representations which are used in turn to influence what the robot perceives.

Neuroanatomy of Multimodal Associative Areas

Investigation of the human being likely started with the study of the body. From the physical and chemical properties of organs scientists could begin to attempt to understand some of their functionalities. When it came to the brain, we were facing a totally different problem: it is not a single organ, but a complex system made by the massive connections between smaller organs units (neurons). To understand its functionality we need to investigate the anatomical structure as well as the process of communication among the network of neurons. It has been suggested that the brain is anatomically connected in a way that is correlated with and facilitates its functional connectivity (Sporns, Tononi et al. 2000).

Cortex is divided into multiple areas which the scientific community more or less agree on, they are defined by their cyto-architecture (type of neurons and other neuronal material composing it), by their connectivity pattern and by the cognitive function they are involved in (Brodmann 1909; Amunts, Schleicher et al. 1999). Areas are connected together, but despite numerous studies, establishing a connectivity matrix is a huge task that has not been achieved yet on human. Historically, the cortex has been thought as being a hierarchy (Felleman and Van Essen 1991); while it is now clear that this is not the case in the mathematical definition of this term (average connectivity rate of 66% (Markov, Ercsey-Ravasz et al. 2010)), a "hierarchical flavor" is still present in our understanding of the early areas connectivity. Studies by Kennedy's team on the monkey provide us with a partial connectivity matrix summarizing which and how areas are connected. Statistical analysis of this matrix gives interesting results: it seems that a general pattern of connectivity exists. Indeed within an area or among areas, the strength of connectivity between two locations seems to be dependent of the distance in the way represented in Figure 1. From the earlier sensory area point of view this organization produces indeed a "hierarchical gradient" of connections to the other areas if we consider that position in the hierarchy is defined by the distance to the sensory cortex. This semi-hierarchical pattern is a well suited design for the multi-modal integration that I will develop in this chapter. After the initial sensory cortex (with V1, A1, S1, G1, O1) where each sensor modality is clearly identified, areas start to be more and more amodal. The premotor cortex of the monkey for example is well known to merge inputs coming from vision and proprioception (Graziano 1999; Maravita, Spence et al. 2003). Merging proprioception with vision is important for biological systems; both

modalities can contribute to a better estimation of the physical body status within the environment, therefore allowing a finer motor control.

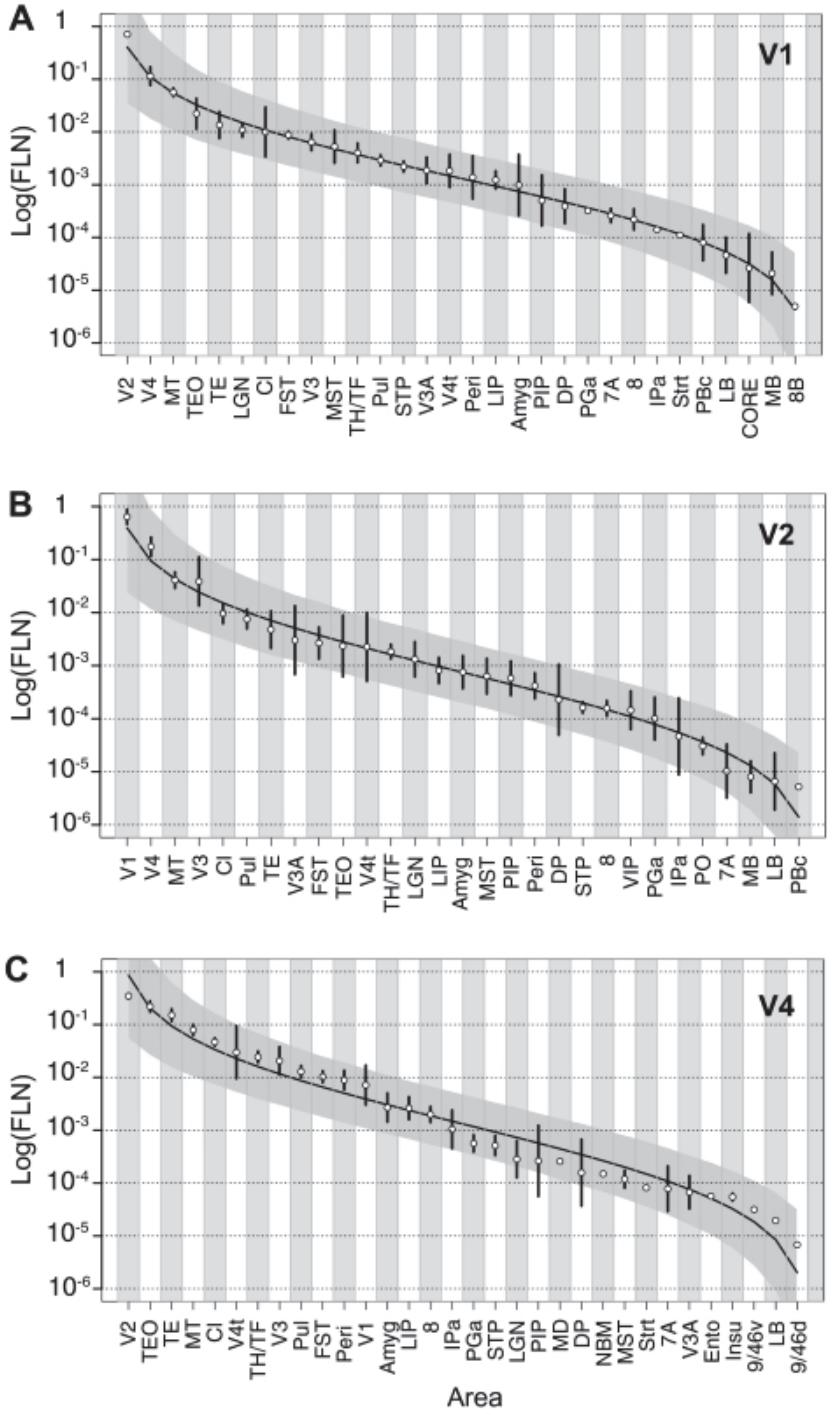


Figure 1: Extracted from (Markov, Misery et al. 2011). The FLN is a measure of connectivity strength. The strength of connection between cortical areas is a matter of their relative distance. Original legend: Lognormal distribution of FLN values. The observed means (points) ordered by magnitude and SDs (error bars) of the logarithm of the FLNe for the cortical areas projecting on injection sites. (*A*) V1 (*n* = 5), (*B*) V2 (*n* = 3), and (*C*) V4 (*n* = 3). The relative variability increases as the size of the projection decreases. Over most of the range, the variability is less than an order of magnitude. The curves are the expected lognormal distribution for an ordered sample of size, *n*, equal to the number of source areas. The gray envelope around each curve indicates the 0.025 and 0.975 quantiles obtained by resampling *n* points from a lognormal distribution 10 000 times and ordering them.

This context leads to the strong intuition that multi-sensory merging is a core principle of cortical computations. When a subject interacts with the physical world, the changes induced are perceived by its sensors, the same action (in the broad sense, any motor act) will produce the same effects, therefore producing a coherence relation between the corresponding sensor activation. When I move my hand in front of my eyes, I will always see it evolving in the same trajectory, with the same shapes, and feel the same exact proprioceptive percepts, i.e. the visual and proprioceptive images are correlated. Since proprioception and vision provide input to the same area, according to the most basic Hebbian rule, visuo-proprioceptive regular patterns will be coded within this multimodal area. This is one of the most obvious relations between our sensory spaces; however it is interesting to look at the case of blind people. Neuroimagery tells us that the dorsal stream which merges proprioception and vision in sighted people seems to merge auditory and proprioception in congenitally blinds (Fiehler and Rösler 2010); while another study shows that early vision during child development shapes definitively the tactile perception of non-congenital blinds (Röder, Rösler et al. 2004). To demonstrate another such combination, visual and auditory signal integration was found in monkey for person identification (face + voice) (Ghazanfar, Maier et al. 2005). Multimodal areas are not predefined to use a specific combination of modalities; they are a mechanism to merge the modalities which express the most pattern co-activation regularity. Listing in an exhaustive way all the multimodal areas and their input would be a huge and meaningless task, even "modal areas" where found to integrate information coming from each other (Cappe and Barone 2005).

Literature about multimodal integration in the brain is vast, and a standalone topic (Meredith and Stein 1986; Lipton, Alvarez et al. 1999; Sommer and Wennekers 2003; Ménard and Frezza-Buet 2005), however the objective here is not to dress a map of the multimodal streams in the brain, but to enlighten the fact that merging of multiple modalities is likely one of the core mechanism induced by cortical connectivity. This principle is the core of the Convergence Zone Theory (Damasio and Damasio 1994) which I will use as a basis for modeling multimodality convergence.

Psychophysics: Illusions

As stated in the introduction, one of the most common and impressive manifestations of the multimodal integration is the perceptual dependency created among different sensory

modalities. It is reasonable to assume that our percepts are based on quite high level areas and do not come directly from the raw sensor input, therefore they encode multimodal traces. From a computational point of view, it means that activity in one modality can produce a form of recall on the other, therefore biasing the perception to a more regular pattern. Most perceptual illusions are indeed inherent to this phenomenon: the ventriloquist and McGurk show the link between auditory and visual percepts (McGurk and MacDonald 1976; Bonath, Noesselt et al. 2007), the rubber hand experiment is about vision, proprioception and touch (Botvinick and Cohen 1998), etc. Taking as an example the rubber hand experiment, in a nutshell the subject is being presented a fake hand as being its own, therefore integrating the fake hand displacement as a displacement of its own limb. Refer to (Botvinick and Cohen 1998; Ehrsson, Spence et al. 2004; Tsakiris and Haggard 2005) for details and variations). In this setup the subject feels a fake hand as being his own, because sensory input coming from proprioception and vision are coherent. The small displacement induced in the vision creates a shift in the proprioception. Indeed given the visual input, the proprioception should not be what the body experiences; the subject therefore feels neither the reality nor the exact vector matching the vision but a mixture of both. In this experiment the illusion is induced after short training, a sort of priming so that the subject can associate the fake hand with his own. Indeed psychophysics demonstrates two types of illusions, one induced by such priming and another related to long term experience of world regularities. While the first shows that multimodal integration is subject to short term adaptation, the second type demonstrates that our experience shapes our perceptual system all along our lives. An entertaining example based on the single visual modality is presented in Figure 2 : the balls seem to be flying or not according to the position of their shadow, while if you hide the shadow they will be on the same level. Knowing that our brain is used to perceive a consistency between the height of an object and the position of its shadow, we can assume that integrative systems is indeed trying to make us perceive the situation in the image as it should be according to the laws of physics. Shadow position and spatial position are so tightly coupled in the world that only manipulating the perception of the shadow induces a major shift in the percept and the estimated position of the object. This illusion is so common and useful that it has been studied (Kersten, Mamassian et al. 1997; Mamassian, Knill et al. 1998) and exploited for artistic purposes. This example is probably not the best one, but the point is easy to grasp: the brain is “fooling us” to perceive not the real world,

but the world shaped as we are used to experience it. Illusions happens when those two worlds do not match, therefore modulating the percept to be a mixture of both.

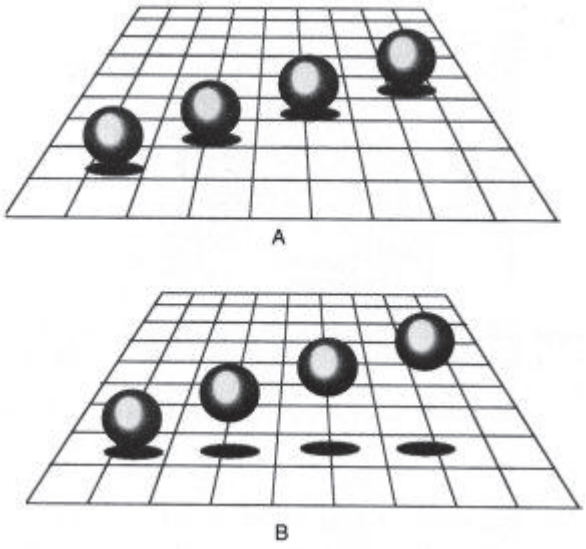


Figure 2: A long term knowledge induced illusion. Our experience of the world shapes our understanding of a perceptual stimulus so that the physics law are consistent with our daily experience. (Picture from Lawrence Cormack)

Indeed illusions do not rely only on early sensory integration; they also touch the "semantic" level with interferences to and from language. Reading "blue" takes longer than reading "blue", and YouTube had quite a buzz about hearing "fake speech" in songs by synchronously reading a text (the illusion might not work for non-native French speakers but one can check in case [4]). This last case is an impressive illusion: despite the "sexual and comic" connotation of the video it is an effect that is worthy of a serious investigation, though to our knowledge no such study has been performed. In the illusion, one reads a text (in French) while listening to a foreign song. The sonority of the text is close to the lyrics of the song, therefore if one reads it a few second before the audio comes, one actually hears what was just read. The auditory percept is "fooled" by vision, but passing via the language level, which therefore demonstrates a three step chain vision->word->audition. Indeed it is tempting to suppose that while learning to read, a multimodal area becomes a convergence zone for written words and their audio representations.

[4] http://youtu.be/w9u4GroWCQY

To conclude this section on illusions, let us consider memories. French literature has a well-known description, by Proust, of what an adult feels when he happens to smell the odor of a cake he used to eat when he was a child. The odor triggers in the reader the “awakening” of dreams where he experiences his childhood, completely disconnected from reality. Memories can be induced on demand, or suggested by environmental factors; however it is clear that in both cases our percepts correspond to an illusion between the multimodal pattern of activation we experienced and the real world. Could we therefore say that the recall process is a “mind induced” illusion, a memory driven activation of a coherent pattern of sensory inputs? This is beyond the scope of this chapter, so we let this question pending and propose a model that can cope with this principle of perceiving each sensory modality shaped by others and by previous experience.

Model: Multi Modal Convergence Maps

The Convergence Zone Framework (CVZ) (Damasio 1989; Damasio and Damasio 1994) makes use of a standard and generic computational mechanism within the cortex: integration of multiple modalities within a single area. This integration derives a memory capability allowing multimodal traces to be recalled using a partial cue (unimodal stimulation for example). The original model formalization was performed by Moll (Moll and Miikkulainen 1997) and is quite similar to Minerva2 (Hintzman 1984) apart the fact that the former uses a neural network while Minerva2 uses "brute force" storage of all the episodic traces. Both models enter the category of Mixture of Experts models (Jacobs, Jordan et al. 1991; Jordan and Jacobs 1994) in which a pool of computational units (experts) are trained to respond to multimodal patterns. When a partial or noisy input signal is presented all the experts examine it and respond with their level of confidence (activation) about this input being their pattern or not. By a linear combination of their responses and their specific pattern the missing or wrong information can be filled in. Another model which can be considered as a special type of Mixture of Experts is the Self Organizing Map (SOM) from Kohonen (Kohonen 1990). While the formalisms are different, the core principle is the same: a pool of neurons is trained so that each of them tunes its receptive field (prototypical vector) in order to be mostly activated by a specific input vector. The SOM is particularly well known because of the direct visually meaningful 2-D map representation, allowing an understanding of the network computation and the possibility to map high dimensional data into a 2D space. They are indeed based on the lateral organization of connectivity within cortical areas, which induces through learning a topographical mapping between the input vector and the neural map. However, despite the fact that they are bidirectional by nature and allow recall, SOMs were never really used as a basis for multimodal integration but mainly to operate vector quantization on high dimensional datasets (Kaski, Kangas et al. 1998). In this section, I will present a model fusing ideas from the CVZ and from SOM. I will first provide preliminary explanation on those two models and finally present the Multi Modal Convergence Maps which I'll link to some very similar models in the recent literature on modeling multimodality.

Convergence Zone & Self Organizing Map

CVZ

A direct model of CVZ has been established by Moll (Moll, Miikkulainen et al. 1994; Moll and Miikkulainen 1997) where multiple modality specific layers are linked through a binding layer (the convergence zone). Each unit of modality vectors is connected toward all neurons of the binding layer with weight being 0 or 1 (connected or not). To store a new pattern, modalities are set and a random pool of binding neurons is chosen, links between input neurons activated and those are set to 1. For retrieval a partial set of the input vectors (e.g. one modality) is activated, the neurons of the binding layer connected with weights of 1 are found and activate back all the input units that they encode for, the process is summarized in Figure 3.

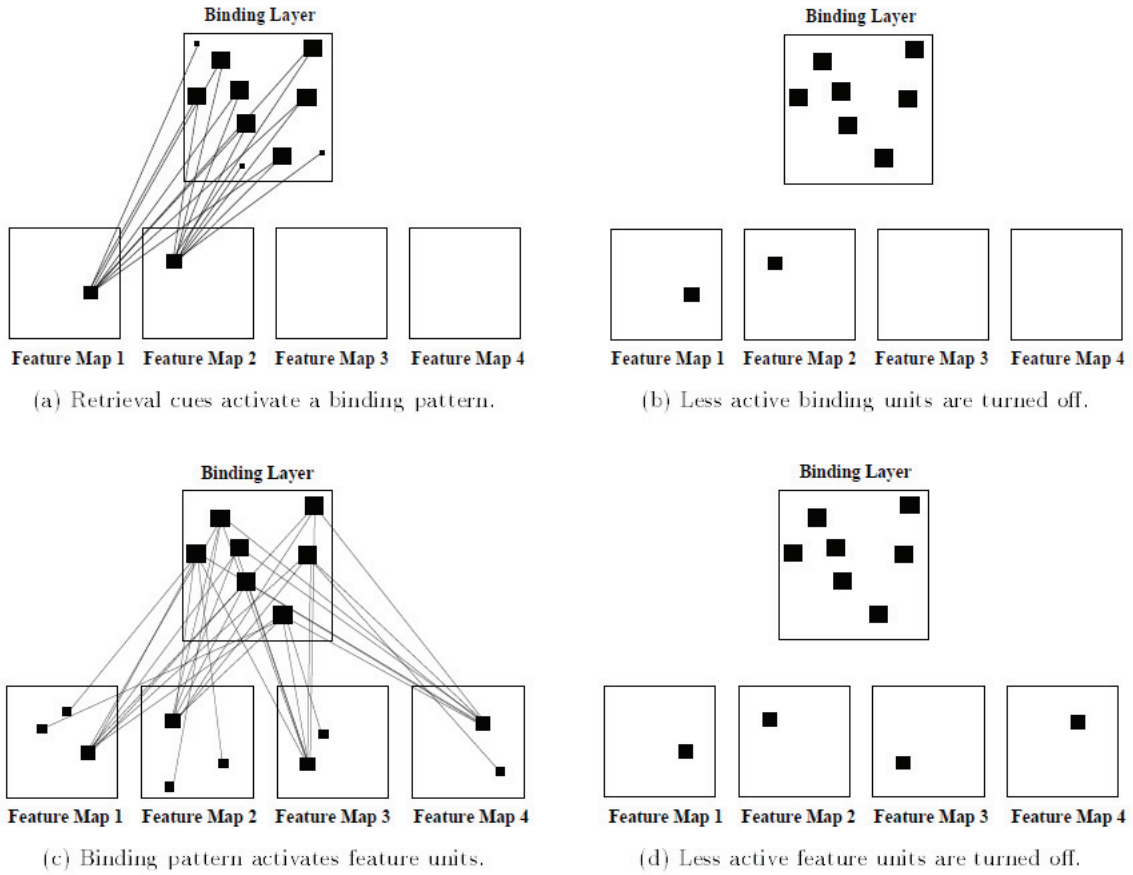


Figure 3 Taken from (Moll and Miikkulainen 1997). A stored pattern is retrieved by presenting a partial representation as a cue. The size of the square indicate the level of activation of the unit.

The focus in Moll's research is to show that such a model can store a large amount of traces within a reasonable number of neurons. Because of this they argue that it is a good

representation of episodic memory storage within the human brain. However, the current interest is more in the properties emerging from the mixture between multiple modalities: how the model behaves in case of incoherent input activity (illusion) and how one modality can cause a drift in the other one. Indeed this model doesn't hold any temporal or spatial relation amongst the stimuli: two stimuli close in time do not have to possess any kind of proximity (while it is typically the case in the real world) and there is no point in trying to find spatial clustering within the binding layer since the position of neurons isn't used at all. Therefore, a multimodal pattern is a hard to imagine "binding constellation" and it is difficult to label those binding neurons to the concept they are related with. Moreover, even if the learning process is fast, there is no evidence about how it behaves against catastrophic forgetting (French 2003) and no benefit from past experience when learning a new trace. The SOM can cope with those points, although it is not designed to handle multiple modalities.

SOM

Self-Organizing Maps were introduced by Kohonen (Kohonen 1990) and have been intensively used and adapted to a huge diversity of problems, see (Kaski, Kangas et al. 1998) for a review.

The main purpose of SOMs is to perform a vector quantization and to represent high dimensional data. A SOM is a 2 (or more) dimension map of artificial neurons. An input vector is connected to the map so that each component of this input vector is connected to each node of the map (see a partial representation of the connections Figure 4). In this context, each neuron of the map owns a vector of connections that has the size of the input vector, and each connection has a weight. The main idea is to fill the input vector with values and to compare these values with the vector of weights of each neuron. Each neuron is activated by the similarity between its weight vector and the input vector. One neuron will more be activated than all the other, we call it the winner.

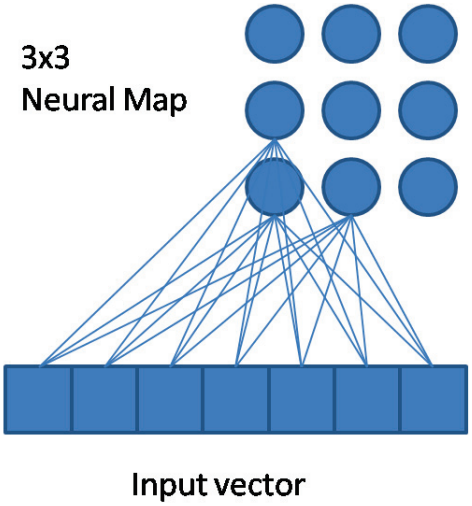


Figure 4: Schematic representation of a self organizing map. A single input vector has each of its neurons connected to each neuron in the map.

The main idea is to train the map so that 2 neurons that are close on the map will encode similar input vectors. To do so input vectors from a training set are presented, the winner neuron is calculated and its weights are adjust so they will be closer to the input values. While this general process is very similar to the CVZ, the learning point is quite important in SOM. Indeed not only the winner neuron is learning, but also its neighbors so that a region instead of a single neuron will learn to respond to this input. The learning rate of neighbors depends of their distance to the winner, this learning function is inspired from the lateral connectivity pattern and the resulting inhibition. The learning rate function is often called the "Mexican hat" because the learning rate is distributed like a sombrero whose center is the position of the winner neuron (Figure 5).

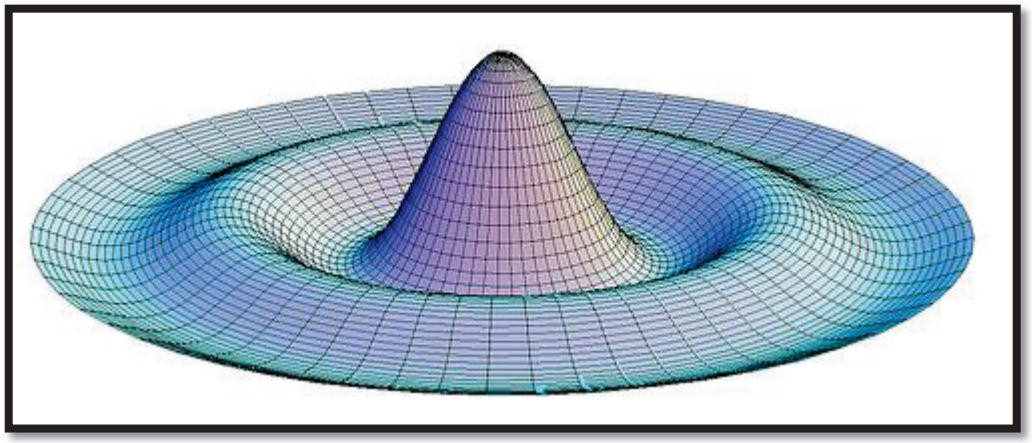


Figure 5: Neighbourhood function or Mexican Hat function (Credit for picture to Daneel Reventlov)

Learning will therefore have a tendency to shape the map so that two similar inputs will be stored within the same region of the map. A concrete example demonstrating this principle is the application of SOM to image compression (Dekker 1994; Pei and Lo 1998). An image is composed of pixels, which hold in most of the cases three channels (R,G,B) accounting for 255*255*255 possible colors. However, when considering a single picture, it is clear that all those colors are not used. By considering each pixel as a 3 component vector and sequentially presenting pixels from an image to a SOM it is easy to get a compact palette of the colors used. Indeed after learning, the map will store gradient of colors in several regions which are composing the most representative palette for this image, therefore the number of color coding the image is the number of neurons forming the map, which can be used to greatly increase the compression. After training the map can be represented by painting each neuron to the color its weights are encoding for, providing meaningful representation and understanding of the map encoding Figure 6.

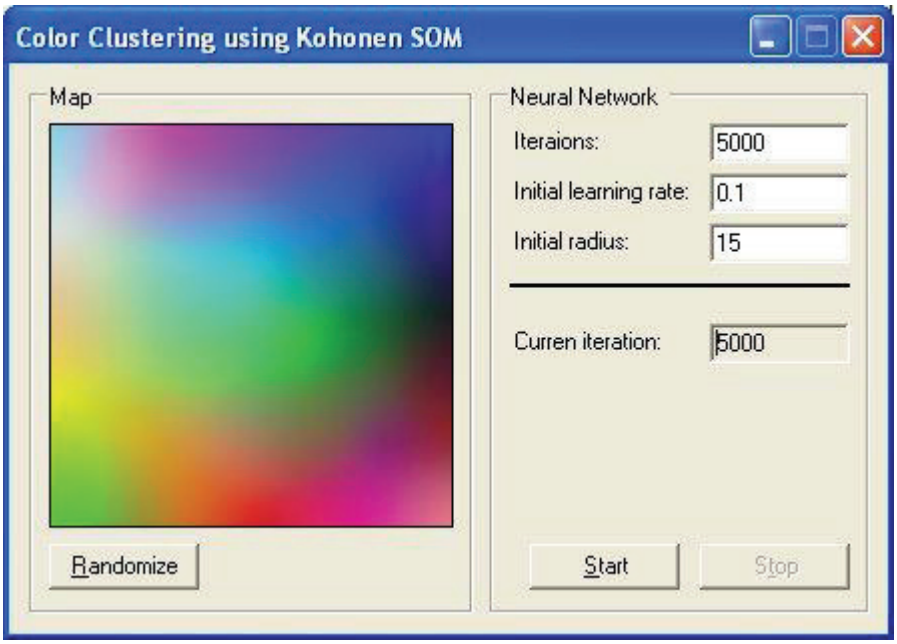


Figure 6: SOM used for color clusterization. Credits to Andrew Kirillov[5] for the demo application.

[5] http://www.aforgenet.com

MMCM

Core Principles

The Convergence zone principle is to store in one place references to multiple lower level activity patterns. It can be seen as a sort of hub, a map of pointers, or more generally as an associative memory. The input patterns are divided into multiple independent units (vectors or maps) depending of the modality they represent, which is something the standard SOM doesn't take into account. On the other side, the initial CVZ model lacks of the self organizing and topographical property inherent to cortical maps. The Multi Modal Convergence Maps are designed to cope with both of these requirements in a unified model merging SOM and CVZ. In a nutshell, it can be seen as a SOM using multiple modalities from which contribution to the network activity can be tuned. A schematic overview of a simple MMCM is presented in Figure 7 and describes in a simplified way the flow of information through the network.

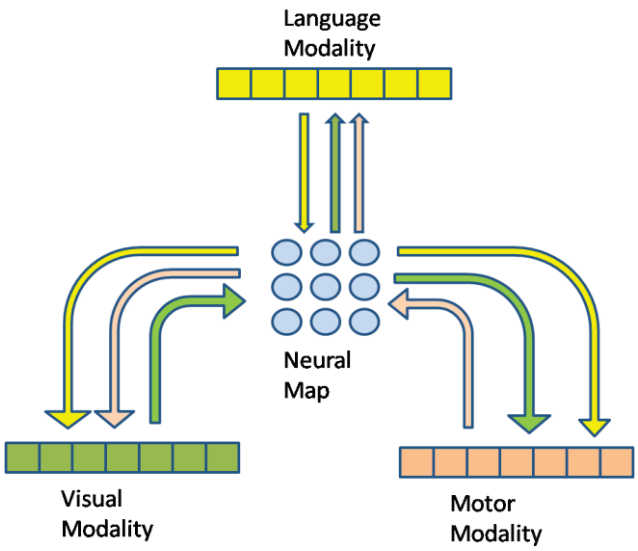


Figure 7: Schematic representation of a MMCM linking three modalities. Each modality is assigned a color, the arrows of the respective colors represent the possible interaction between modalities created by the convergence map.

Each modality is taken into account during the map activity calculation according to the equation (1) & (2), with P_i^m being the ith component of the input vector perceived by modality m and I^m the Influence factor of the modality m. Modality influence is a number in [0,1] which represents how much a modality contributes to the map activity, in comparison to the others. In our implementation, the "map" is in fact a cube, neurons are distributed along 3 dimensions which means that A_{xyz}represents the activity of the (x,y) neuron of the

layer z. This third dimension idea came from the fact that the cortical maps are divided in 6 layers (Brodmann 1909); while this has not been extensively examined, we can consider that adding a dimension allows a higher storage capacity by providing more nonlinear transformations to be represented while keeping the topographical properties of such representations. We will come back to this point in the parameters explanation, but to give the main idea the third dimension seems to increase the encoding potential of the network.

$$a^m = I^m \times \frac{\sum_{0\to n}^{i} \left| W_{ixyz}^m - P_i^m \right|}{n} \quad (1)$$

$$A_{xyz} = \frac{\sum_m a^m}{\sum_m I^m} \quad (2)$$

To ground the discussion in reality, consider that a robot is looking at its hand, which is changing postures, and listening to an observer say the names of these postures. If we take the map from Figure 7, at each step three vectors are obtained from the robot sensors: the image (visual modality), the joint encoders (motor modality, similar to proprioception) and the words recognized by the spoken interaction (language modality). All the respective modalities inputs are activated according to these vectors, and then the map activity is calculated. The most activated neuron of the map (i.e. the winner) is recorded and its weights give the prediction for each modality. If the learning mode is on, the weights of each neuron in the map are adjusted according to the equations (3), (4) and (5).[6]

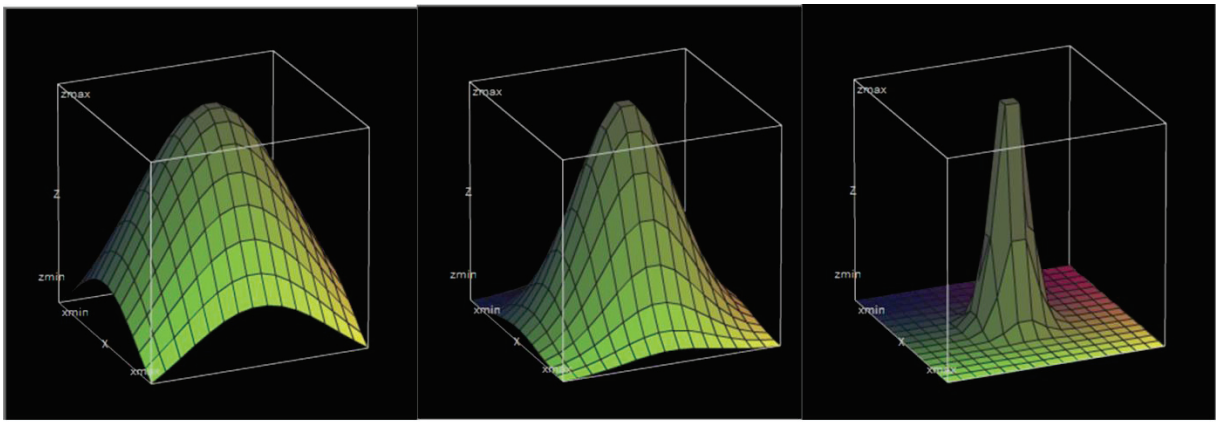
$$D_{xyz} = \frac{1}{\sqrt{2\pi}} e^{\frac{-((x-x_{win})^2+(y-y_{win})^2+(z-z_{win})^2)}{2\sigma^2}} \quad (3)$$

$$E_{ixyz} = W_{ixyz}^m - P_i^m \quad (4)$$

[6] Note that the modalities influence is not taken into account during the learning, at the time of writing this function is not implemented, however it could be an interesting way to guide learning. For example, it could be a model for what happens in congenitally blinds (influence of vision during learning 0) who are given back the sight by technological means (influence of 1 during perception) and experience meaningless percepts.
For more info about artificial retinas see: Dobelle, W. H. (2000). "Artificial vision for the blind by connecting a television camera to the visual cortex." ASAIO journal **46**(1): 3.
, Humayun, M. S., J. D. Weiland, et al. (2003). "Visual perception in a blind subject with a chronic microelectronic retinal prosthesis." Vision research **43**(24): 2573-2581.

$$dW_{ixyz} = \lambda . D_{xyz} . E_{ixyz} \quad (5)$$

To summarize the equations, the winner weights are modified to better match all the input vectors so that next time those inputs are presented this neuron activity will be higher. The other neurons of the maps also learn with a rate depending on the distance separating them from the winner on the map. Figure 8 represents the distance function presented in equation (3) with various values for sigma, we could also have used a Mexican Hat presented in Figure 5, the main requirement is that the function represent a decreasing gradient from the winner node to its far neighborhood.



Sigma = 0.5 Sigma = 0.3 Sigma = 0.1

Figure 8: Neighborhood function used in the MMCM Library, accounting for only 2 dimensions and different values of sigma (represented for x and y ranging from -1 to 1). The Z axis represent the learning rate, the center of the dome is the winner.

This neighborhood learning creates the self-organization of the map: inputs that are similar will activate close regions on the map. At the modalities level, the input vectors are being classified in regions on the map, for example all visual pattern of the hand seen from the back will be stored on the top right corner, while the hand seen from the front will be on the other side of the map. Since each modality will have a tendency to create its own regions resulting from clustering of the inputs, the map will organize itself in multiple superposed partitions. This arrangement can be seen as an associative mapping linking vectors of different modalities that are often sensed together: assuming that we train the MMCM of Figure 7, the robot is looking at its hand while sensing it and feeling it occurring in consistent activation of the map coming from vision and proprioception as described in Figure 9. For more explanation about this figure and the associated experiment, please consult the experimental results section of this chapter; the most important thing to understand at this

point is that the convergence map creates a spatial organization of regularities extracted from the sensed modalities.

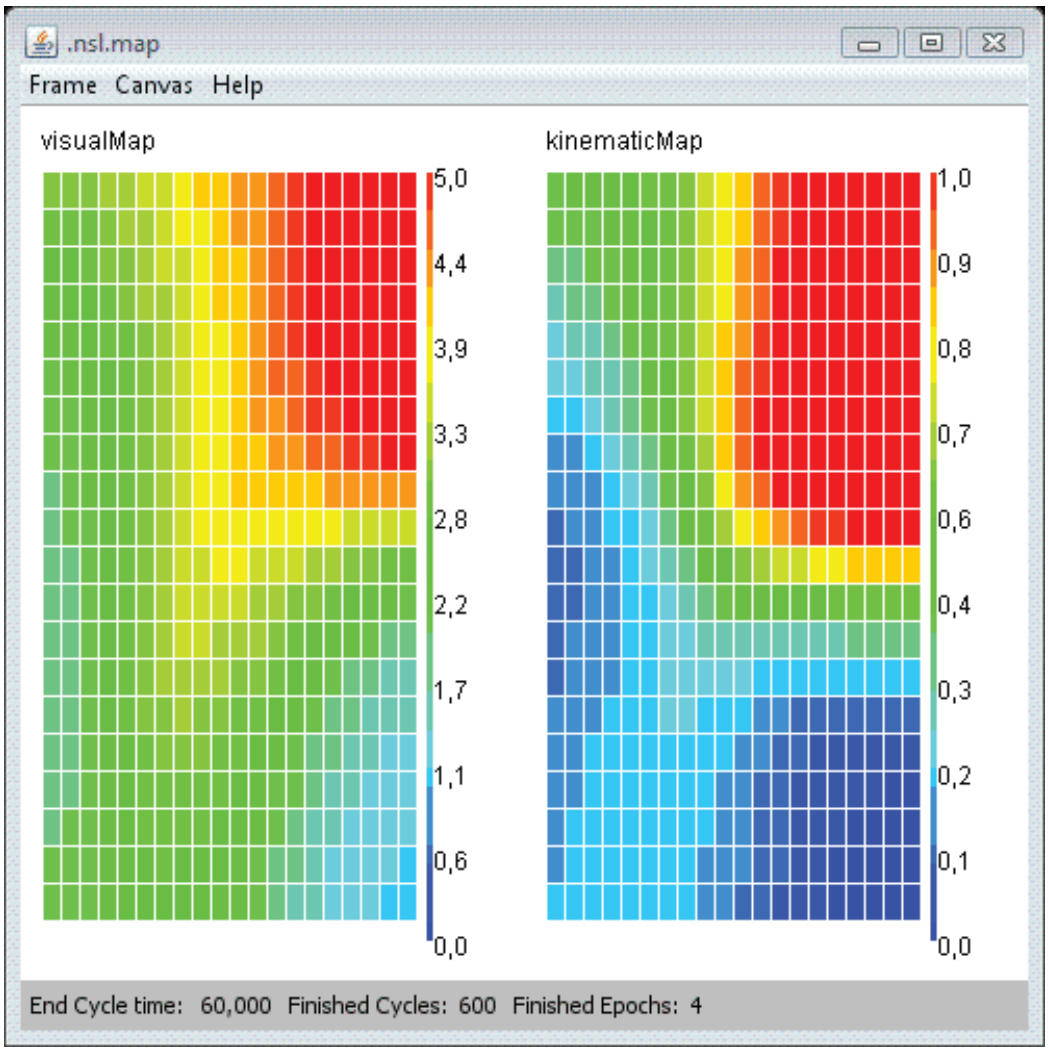


Figure 9: Visualization of the map activity when stimulated by vision (left) or proprioception (right). The most active spot is the same for both modalities, because vision is consistent with proprioception. In the case of an illusion those two spots would be different and the percept would be a mean of both.

Enactive Perception: shaping the world

The word "enaction" will be employed to explain a psychophysical phenomenon and a mechanism of the MMCM model. This word refers to a well-specified philosophical concept (Maturana and Varela 1987), however here we will use only a part of it: the fact that physical experience of the world shapes perception. We will not consider high level considerations such as mind, consciousness and the link between body and mind, but instead focus on very low level perception. Even prior to action, though actions are tightly linked to perceptions

since they modify them, let us consider the pure perceptual question. How does the brain treat the information coming from the sensors to produce a percept, and to what extent is this percept actually different from the raw sensory information? A major point of the enaction concept is that we perceive the world from our point of view, and this point of view is biased by the previous experience we had of the world. Enaction is a concept that involves a whole species and that is developed through evolution, although we can apply the same idea at the individual scale. Through its life an individual experiences the world, acts on it, perceives it, and regularities are extracted. The brain is shaped to perceive this world as it appears every day, its weights are modified so that the position of a shadow and the respective height of its owner object are interdependent (Figure 2), because this is a physical law inherent to our universe. This basic knowledge that we are not even conscious of possessing is so deeply trusted that it can produce the illusions mentioned above and in Figure 2. We do not perceive the raw information coming from our various sensors as it comes, instead our brain always attempts to make it close to a phenomenon we are used to experiencing, it tries to shape our perception of the world so it looks like the world we know. This is what I call Enactive Perception, the fact that our brain is always mixing the reality with the archetype of reality it possesses, while at the same time building this archetype to cope better with its new experience. The benefit of such a shaping of our perception by our knowledge is not trivial; however it is an undeniable aspect of our cognition which is likely to be due to the feedback stream connecting most of the cortical areas with lower areas of the hierarchy. According to anatomical results, convergence zones, if they exist, are not different from others areas, therefore they are sending feedback information to the modalities they are fed by. The MMCM model take this process into account by considering that a perception (either modal or amodal) is the result of merging a “real stimulus” and the associated prediction of this stimulus.

Using the MMCM it is possible to link multiple different modalities according to the way we experienced them. It is possible to predict one modality from the activation of the map. We use this prediction capability to embed the MMCM into a dynamic model which can represent the enaction. The main idea is that the map doesn’t perceive directly the sensory data (reality), but the existing map memory altered by those sensory inputs. Indeed there is growing feeling in neuroscience field that our brain computation are mainly “internal” and

only driven by the input coming from our sensory system (Raichle, MacLeod et al. 2001). The Figure 10 represents schematically how the MMCM model takes into account the enactive perception.

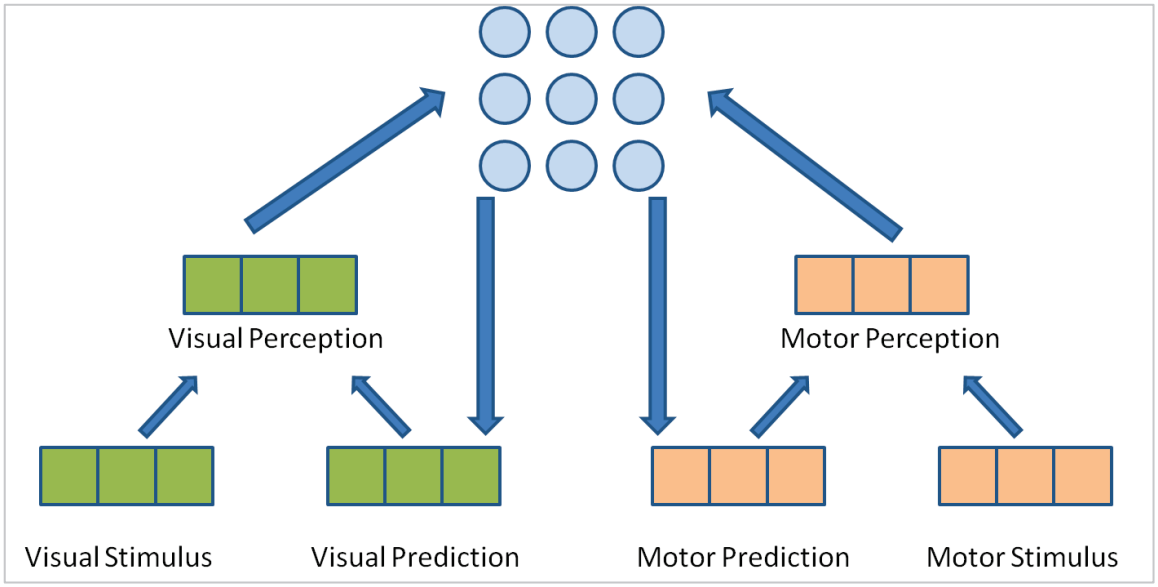


Figure 10: Enactive MMCM. The map does not perceive the input directly, but it uses a mixture of real and predicted inputs. This process adds a temporal sensitivity to the system.

At any time the system experience the world, so the stimuli vectors are set by the sensory input (St). However, this raw sensory data is not used alone to stimulate the MMCM, instead we use a mixture of this sensory input and the prediction (Pr) made by the system at the previous time step. This perceived vector (Pe) is calculated using the equation 6.

$$Pe_t = \rho S_t + (1 - \rho) Pr_{t-1} \qquad (6)$$

Rho is a parameter of the system which vary in [0,1] and represents the weight of the reality in the perception of the system. The smaller it is the more the system will be influenced by its memory. Adding this prediction to the inputs gives the ability to the MMCM not only to act as a static associative memory, but also to process temporal information. This means that the system should be able to learn sequences of stimuli; while it has still to be tested and could be very much dependent from the value of Rho (if we set it to 1 for example the system loses completely this capability.)

Parameter Influences

The MMCM possesses several parameters that can have a major influence on the model behavior and computation achieved. In this last part we consider all those parameters and provide information about their role and the way they are intended to be tuned.

Modality Influences

Each sensory modality has an influence on the perception which tells how much it "drives" the percept in comparison to other modalities. While some research has attempted to quantify those influences in the human cognition (Burns, Razzaque et al. 2005), it is likely the "weight" of each modality is not absolute but depends of the task, the subject and his will. In the model, a number between 0 and 1 is associated to each modality and represent its level of contribution to the final percept. We can see in equations (3) and (4) that we are operating a sort of softmax function to calculate to which proportion each modality contributes to the final activation of the map. An influence of 0 means that the modality is not taken into account at all, for a vision based example it would mean that I close my eyes. Setting a modality influence to 0 can be useful if I want to make predictions of those modality based purely on the other; for example I could predict what my hand would look like based solely on proprioceptive cues. Of course it is possible to set influences to intermediate values (for example vision to 1 and proprioception to 0.5) in order to test hypothesis on the psychophysics results or just to "trust more" one sense in comparison to the others.

Enaction Factor

The enaction factor, as I described before represents to what extent the previous knowledge contributes to the perception of reality. Perception is a mixture of what our brain predicts about what we should perceive and what really comes from our sensors. However, while the model is quite adapted to represent static associative memory, using its capability to be enactive turns it into a temporal model. Indeed the map makes prediction for each modality based on the sum of all perceptions which have an influence superior to 0. If all modalities are taken into account it means that the perception is dependent on itself. To cope with this problem we consider that perceptions in the world are continuous through time. What our eyes see won't be radically different from one image to another: olfactive and tactile

information doesn't "jump" from state to the other, there is a sort of perceptive continuum inherent to the physical world. Therefore, we can consider that the percept experienced at time *t* is very much related to the percept we experienced at time *t-1*. Given that assumption, the model defines the perception at time *t* as a mixture between the real modality signal (coming from the sensor) at time t and the prediction made for this modality at time *t-1*. The enaction factor is a number ranging from 0 to 1, with 0 meaning that the previous prediction is not taken into account at all and 1 meaning that only the last prediction is considered, therefore running the system in a "reverberative state" with no influence at all from the reality. Such a system has a strong analogy with dreaming: we used to say that "feeling cold makes you dream of snow", indeed with an enaction factor close to 1 the reality just has a driving effect on the perception, therefore leaving the system in its self-induced state. With non-extreme values, the enaction factor can be a way to break or create illusions (which are basically occurring because our brain trusts more what it used to know that what it actually sees).

Learning Rate

The learning rate, as in every learning machine system, is a key factor. It represents the amount of modification applied to the weights of the winner neuron during learning. It is ranging from 0 to 1 with 1 meaning that the weights are adjusted in one shot so that the winner neuron encoded vector will match exactly the input presented. We never want this to happen; instead a smooth modification of weights should be applied. Moreover the learning occurs on the winner neurons but also in its neighborhood, meaning that a too high learning rate can easily destroy previous knowledge stored by the map. However, a too low value would induce a very slow learning. Indeed all the generic problems of learning rate in neural networks apply to the MMCM model and it is likely that the best (but not yet implemented) solution would be an adaptive learning rate based the prediction quality (the difference between prediction of the network and the real sensor input). Considerations about adaptive learning rate and how to tune it have been investigated and are not the focus in the current study, see (Jacobs 1988; Magoulas, Vrahatis et al. 1999; Plagianakos, Magoulas et al. 2001) for references.

Sigma

Sigma is meaningless for the size of the neighborhood affected by learning. As depicted on Figure 8, the closer a neuron is from the winner, the higher its learning rate will be, according to a Gaussian like function. Sigma has no "absolute" value and should be chosen according the size of the convergence map used. In many SOM algorithms, it is set initially to encompass nearly all the map and it decreases over time so that the clustering becomes finer. Indeed the adaptation of sigma is a critical parameter for all models of the mixture of experts type. For example, in the model MOSAIC (Kawato 1999; Lallée, Diard et al. 2009) the value of sigma is "hand tuned" over the course of the simulation in order to allow proper learning. However hand tuning cannot be a solution, Sigma must be set in an automatic way by the system. A good prediction should have a smaller neighborhood in order to refine the learning and to tackle catastrophic forgetting. Typically the neighborhood range should be decreasing while prediction quality increases; this would allow initially the whole map to be shaped by global regularities, while detail learning would be encoded at the local level. However, it supposes that the training samples are presented "homogeneously". A good work around is to define a large neighborhood while keeping the learning rate quite low, this way all the map learns, but the adaptation is slow and does not overwrite existing knowledge. However, probably that the best solution would be to base the calculation of sigma on the prediction quality and the "possibility to learn" in a similar way to what is achieved on intrinsic motivation to learn by Kaplan (Oudeyer, Kaplan et al. 2007). Their idea is that the agent should choose the action which allows him to learn the most. Not the one from which he can predict flawlessly the consequences, nor the one that he is unable to understand at all, but the one which he understand enough to verify its prediction. Similarly the learning rate and size of neighborhood in MMCM and mixture of expert models could be low for very poor and very high prediction quality, while the average predictability should be assigned much higher values.

Similar Models

Robotics and computer science in general is thought of as permanently "reinventing the wheel". The same concepts are rediscovered again and again, systems with similar purposes are re-engineered and we often decide to do by ourselves something that has already been done by others. However, science is about understanding, about grasping what is really

behind the system and this cannot be achieved just by reading and accepting the word of others. The MMCM model was motivated by historical bibliography, including the CVZ and SOM models and its design derives from those bases. Indeed these were the only necessary and the most relevant material to the problem of multimodal convergence. Afterwards, I discovered the Multimodal Self Organizing Map (Papliński and Gustafsson 2005; Papliński and Gustafsson 2006) which is very similar to the MMCM. The principal difference stands in the integration of feedback (enactive perception in my case) and the impossibility to modulate separately the influence of each modality. Moreover, while the model has been tested extensively on a theoretical sample case (classifying animals based on modalities coding for their attributes) it hasn't been applied to real modeling of multimodal sensory convergence in an embodied robotic framework. However, I have no doubt about the capability of this model to cope with a robotics implementation. Another self-organizing multimodal model close to the MMCM is being investigated by Mathieu Lefort (Lefort, Boniface et al. 2010; Lefort, Boniface et al. 2010; Lefort, Boniface et al. 2011), it focuses more on low level mimicking of neural process with the modeling of cortical columns. Although this model also seems very well suited to robotics and embodied multimodal integration, it hasn't been applied to this topic so far.

Despite that this field of research is quite small, there is an increasing interest in modeling cortical associative maps within the convergence zone framework. Multimodal association can be modeled using various mathematical tools, some of them probably more efficient or formalizable than MMCM. However a strong point of SOM and MMCM is the ease of understanding the ongoing process and the relatively intuitive functioning of the multimodal association. Moreover, MMCM includes core specificities like the enactive perception or the independent modalities influences which make it a unique tool to model several psychophysics results observed on human.

Experiments

One of the principal multimodal integration domains studied in primates is that which merges vision and proprioception in the context of grasping. In many body configurations, the hands of a subject are within his visual field, therefore making them the best candidates for the integration of those two modalities (vision and proprioception) which are indeed tightly linked for those limbs. With eyes closed and the hand moving, it should thus be

possible to get a fairly good mental image of the appearance of the hand just based on the body feeling, although the reverse operation needs a bit more of experimental setup to be investigated, it is also clear that vision contributes to proprioception. The following experiments describe various ways to model, demonstrate and use this integration within embodied framework of the iCub robot.

Conducted: Proprioception enhance vision speed

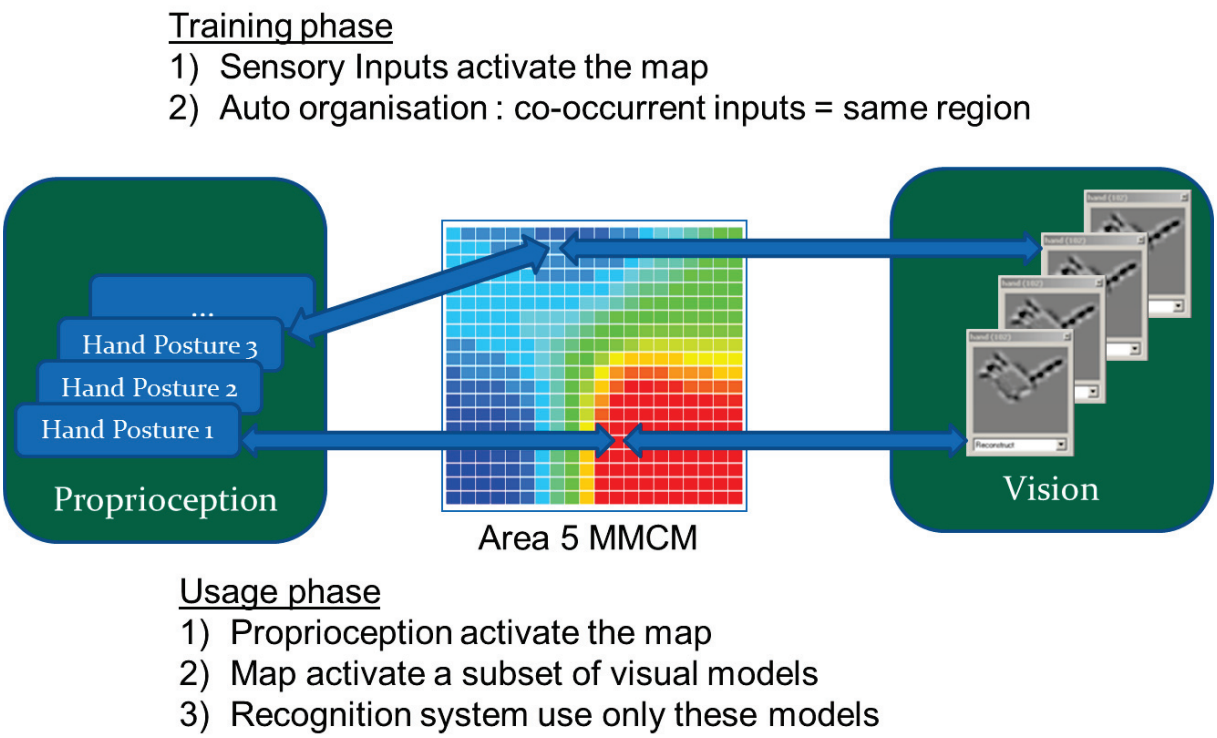


Figure 11: Summary of the experimental process.

The main application of MMCM on a real robotic platform has been achieved on the iCub (both simulator and real robot) and results have been presented in (Lallée, Metta et al. 2009). Our goal was to efficiently grasp objects recognized using vision; the main problem in grasping at this time was the inconsistency between the coordinates of an object obtained through vision and the position of the hand when commanding its Cartesian controller to move to reach this point. Due to minute errors in calibration, those two positions where not identical, therefore resulting in a hand displaced relative to the target of the reach (see Figure 12) and the robot failing to grasp. The solution found was to proceed to an initial reach of the object, visually detect the hand and the target, calculate the difference and reduce it by repeating this process in a closed loop. However, the hand is a deformable

object: according to its kinematic configuration it can correspond to a functionally infinite space of different visual appearances, thus rendering the recognition problem not-trivial.

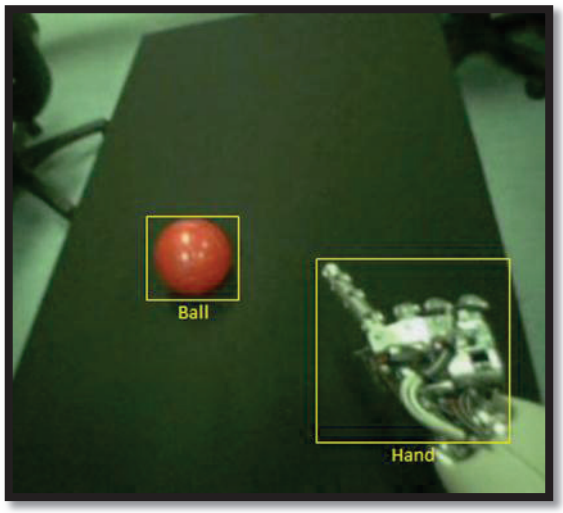


Figure 12: Status of target and hand after the initial grasp of the iCub. The distance between the hand and the ball needs to be reduced using a closed loop (error reducing) control. Visual recognition is achieved using Spikenet (Thorpe, Guyonneau et al. 2004).

The visual system of the robot is based on a robust pattern matching system (Spikenet (Thorpe, Guyonneau et al. 2004)), which means that an object was visually defined as database or set of models (patterns) which were extracted from images of the object. In order for an object to be recognized it should be modeled from several view points and in all possible configurations, which results for the hand in the creation of an extensive number of models (see Figure 13). Of course, the performance of such a system in term of recognition time depends mainly on the number of models it is asked to check for: in the case of the hand the system became intractable.

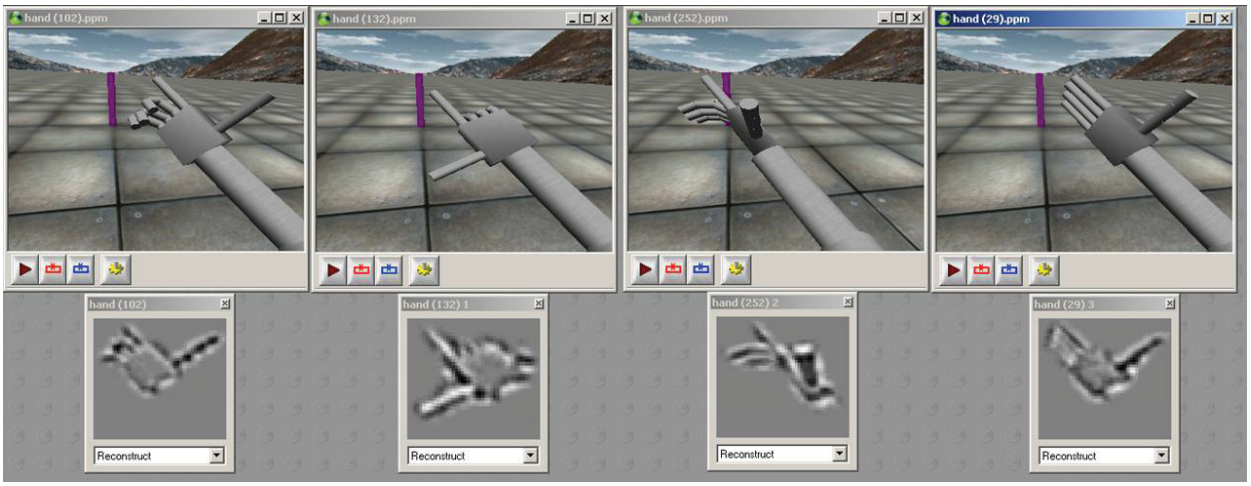


Figure 13: iCub visual models of the hand in a few configurations. The visual pattern changes dramatically from on configuration to the other and a huge amount of models is needed to recognize the hand in every posture.

However, not all the models are relevant in every situation: since the recognized item (the hand) belongs to the robot, it is possible to take advantage of the embodiment information in order to reduce the complexity of the recognition process. Indeed, given a kinematic configuration, or a proprioceptive vector, the model database can be reduced to a subset of relevant models. The MMCM was used to identify this subset: a map linking vision and proprioceptive modalities was built, in the following manner.

The robot gazed forward, and with its hand in its visual field, rotated the hand about the wrist while opening and closing the fist. Proprioceptive signals were collected from the joint angle sensors, and visual signals from the vision recognition system. The vision modality was a vector of M components, M being the size of the full database of hand models. At each time step, the visual modality was obtained by setting the units corresponding to recognized models to 1 and all the other to 0. The proprioceptive modality was a vector of 16 components corresponding to the encoders of the robot arm scaled between on [0,1]. The experiment was divided in two phases 1-Learning, 2-Recognition. During the learning phase, the robot was looking at its hand while moving it in a semi-babbling mode as depicted in Figure 14. The full model database was loaded in the vision system, occurring in a slow recognition, and both visual and proprioceptive modalities were feeding the convergence map. The convergence map, MMCM, learnt to associate a kinematic arm configuration with its subset of activated models in approximately 8 minutes of babbling.

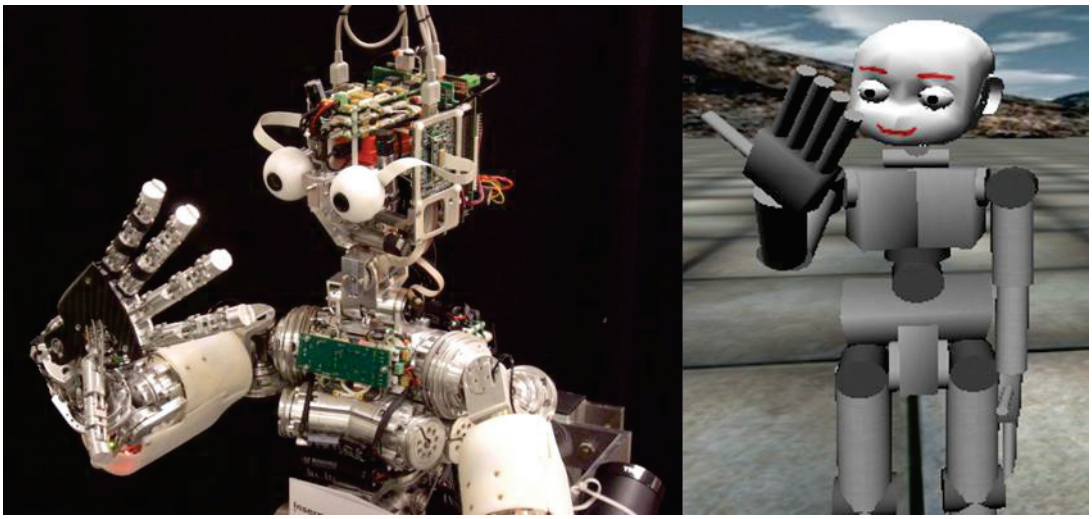


Figure 14: iCub robot and simulator learning to visually recognize their hand based on their proprioception.

Once the map has learned, its predictive capabilities can be used. The influence of the vision modality is set to 0 so that the map gets its activation only from the proprioception. At each time step, the proprioception is sensed and the vision vector is predicted therefore producing the subset of models which should be recognized in this configuration. The visual system restricts the database of recognizable models this subset in order to allow a faster recognition than if it was using the whole database. The effects of this pre-selection of visual patterns are presented on Figure 15.

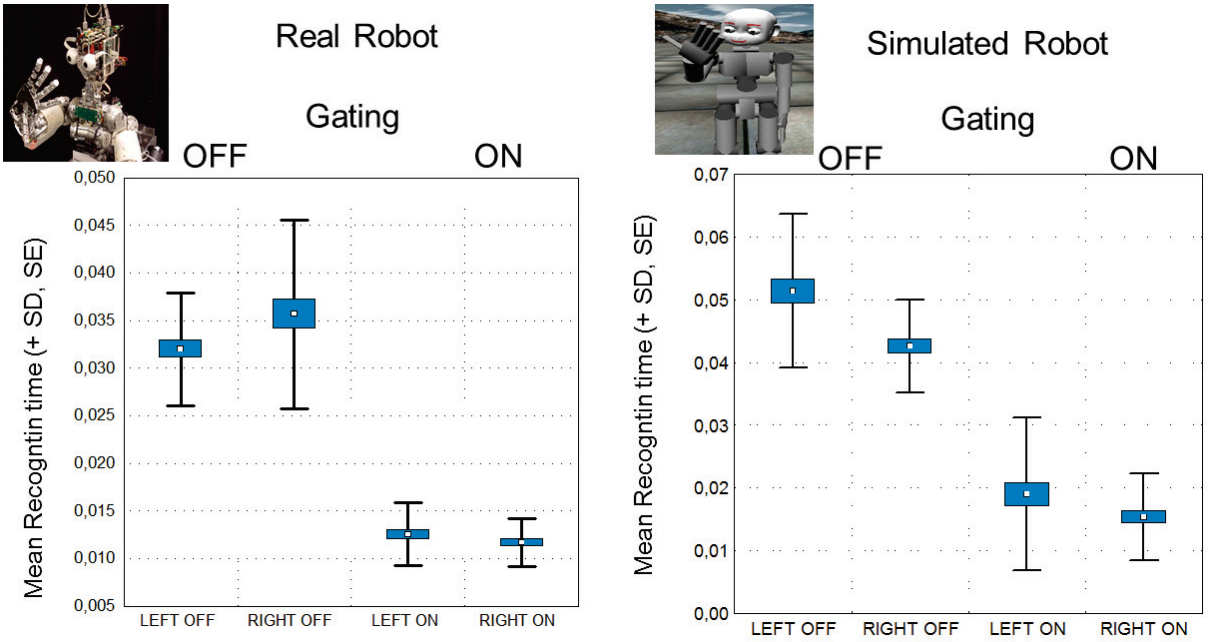


Figure 15: Effects of proprioceptive gating on visual recognition time. Experiment was conducted twice: once on the simulator and once on the real robot, similar highly significant reduction of recognition time was found. Gating is the reduction of the set of recognition candidates, by the predicted candidates from the MMCM based on the proprioceptive position of the hand.

The effect of this pre-selection is of course dependent of the number of models present within the database, in our conditions.

To be conducted

Although the MMCM library software developed in the thesis provides everything needed for the design of many experiments, time constraints and focus on higher level cognition didn't allow them to occur. However, a few of them were (and are still) planned with the collaboration of Alessandro Farnè who has conducted research on integration of proprioception and vision in the human. As mentioned above, psychophysics provides us a lot of insights on the interferences between modalities. In this section I will present two experiments that have been designed to test those hypotheses although they haven't been conducted at the time of writing.

Experiment 1: The rubber hand experiment

One of the most famous and studied experiments in vision and proprioception is the rubber hand. In this experiment the subject is habituated to see a rubber hand which is not his own while tactile contacts are done synchronously on the fake and real hand. This way both vision and tactile information are congruent, encouraging the subject to feel the rubber hand as his own. The matching between modalities allows bypassing the "plastic" aspect of the hand: if I see something touching the hand and I feel it at the same time, then it must be my hand. After this training phase, if the hand is presented shifted, then motor commands of the subject will be impaired based on this displacement. This effect has been studied both from psychophysics and neurophysiology sides (Botvinick and Cohen 1998; Ehrsson, Spence et al. 2004; Ehrsson, Holmes et al. 2005; Tsakiris and Haggard 2005) and are often targeted at finding how body ownership feeling is handled by the brain.

The hand is a part of the body that one perceives from his birth using three modalities: vision, proprioception and touch. The signals coming from those sensors are linked within a convergence zone (more likely a network of areas (Maravita, Spence et al. 2003)) which is able to learn regularities in the relative relations between those senses. We have seen in the experiment conducted on the iCub that proprioception and vision of the hand are directly related; extending this, in the case of touch a contact will be detected both on the visual percept and in a tactile way. We can model this three modality convergence using a MMCM

therefore allowing to replicate the rubber hand experiment on the robot and retrieve the shift of perception directly in the network activity. Detailed setup of the modalities coding is presented in Figure 16 and assumes some simplifications on the visual and tactile components.

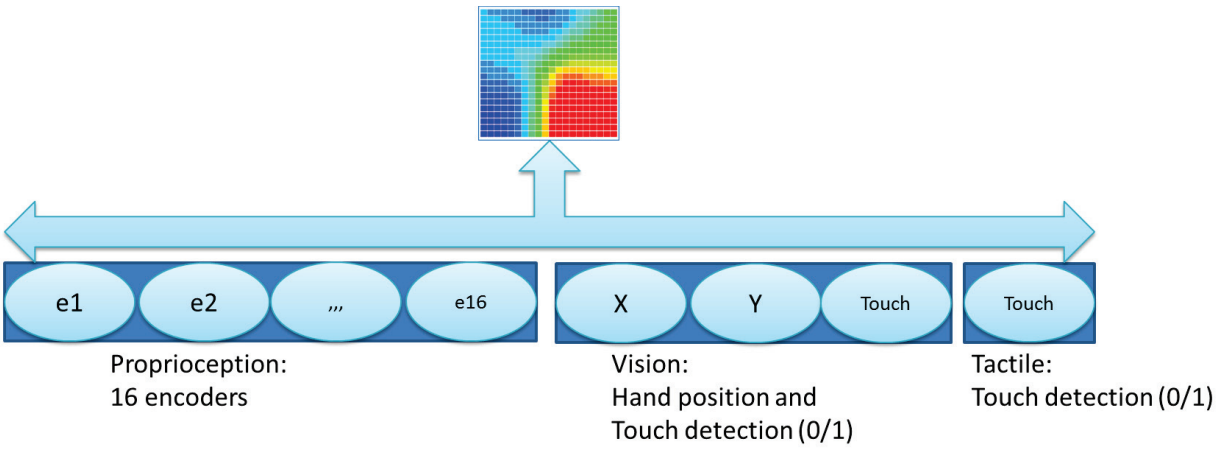


Figure 16: Modality coding for the rubber hand experiment simulation.

Vision represents the position of the hand in the visual field and a Boolean value describing whether there is a contact between the hand and something else. The tactile part receives a Boolean which is the result of a contact detection using the iCub skin. Finally proprioception is the vector representing each joint angle of the robot arm.

The rubber hand experiment in human needs to fool the visual part by using a fake hand. With our robotic design we could reproduce the experiment by directly cheating on the perception of the robot. For example we can add to the sensed hand position an offset which would act as the displacement of the fake hand in human. Moreover other perceptions could be modified this way: we could probably show that a disturbance in the proprioceptive feeling produces a shift in the visual localization of the hand.

In order to make the experiment closer to that on humans, we could also add another visual component which would code for the "visual similarity" of the hand. We could play on this parameter to give testable predictions on human in order to further validate the model.

Experiment 2: Rock, Paper, Scissor experiment

Another experiment could add a flavor of "reinforcement learning" to the MMCM model. Assuming we would like to teach the rules of the "rock, paper and scissor" game to the robot

which means, given a sign, which one should be used in order to win. Using exactly the same map pattern as in the previous experiment (same modality sizes) and the coding presented in Figure 17, we can implicitly teach the game's rules by demonstrating the game. The learning phase would unfold as:

Repeat:

1. Human moves in either rock, paper or scissor posture.
2. Vision modality is set accordingly, Reward is set to 1 and proprioception is predicted.
3. The robot posture (rock, paper, scissor) which is the closest from this prediction is commanded.
4. The human says to the robot either "you loose" or "you win".
5. Proprioception is set to the played posture, vision is set to the human move and reward is set to the result of the game. The map learns.

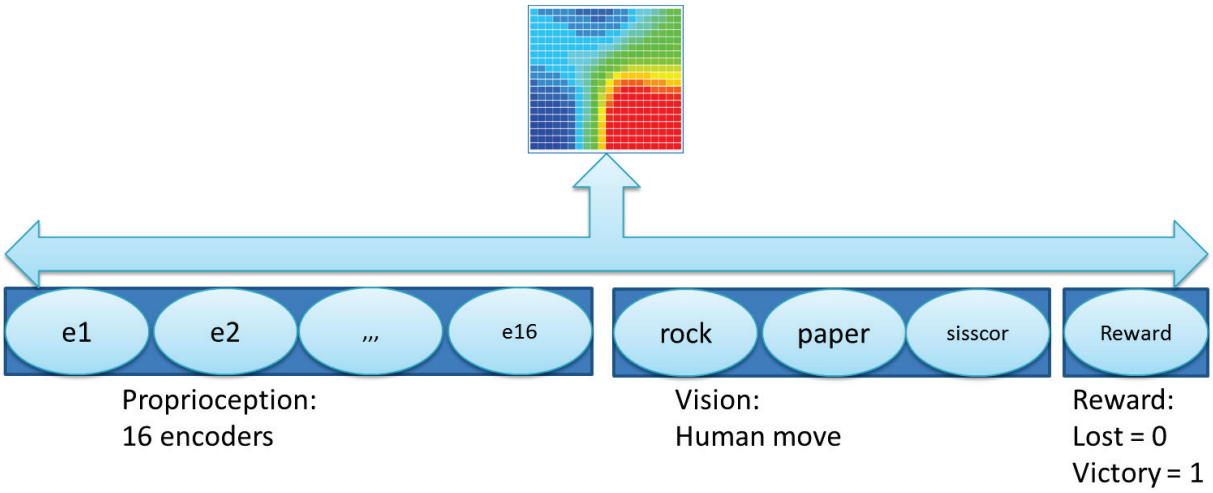


Figure 17: Modality coding for learning the Rock, Paper, Scissor game. After the map has learnt, the rules of the game are coded within it.

After a few rounds the robot should start to do the move which should beat the human. Because we ask the robot move (proprioception) based on what the human did and an intention to win the game (reward is set to 1 before the prediction). The omnidirectionality of the map allows also some other use: say if the robot won the game based on his move and the human one, or predict what the human would have to play in order to win/loose the

game. This experiment would be less related to human data, indeed its main purpose is to show that MMCM model can be used to model rules and logic operations.

Due to the possibility to set modalities influence to 0 in order to make predictions, MMCM model is more than a simple auto associative memory. Indeed it can also be used as an hetero associative memory and therefore be used to model functions instead of only cue based pattern retrieval.

Discussion

Multimodal fusion is a core principle of cortical computation. It can serve as a basic principle to explain many behavioral results and as a source of inspiration for the emergence of concepts from sensorial data. The MMCM model has been designed to reproduce this behavioral data and turns out to be generic enough to cope with a quite large range of problems. A detailed description of the software produced (MMCMLib) is provided as an Annex, however a few notes on this. It uses the YARP library, which allows message passing over the network. Each modality can be set remotely, and its prediction can be read as well. This mechanism is fairly important in the case of a hierarchy of maps. The current work does not address using the MMCM map as a building block for more complex hierarchical networks, however the software has been designed so that one map can serve as a modality for another. By allowing maps to be connected remotely, the hierarchy processing can be easily parallelized over multiple computers, therefore solving computational issues which may occur. Future work will focus on achieving the experiments presented in this chapter, however it is quite appealing to imagine more complex processes like the one depicted in Figure 18.

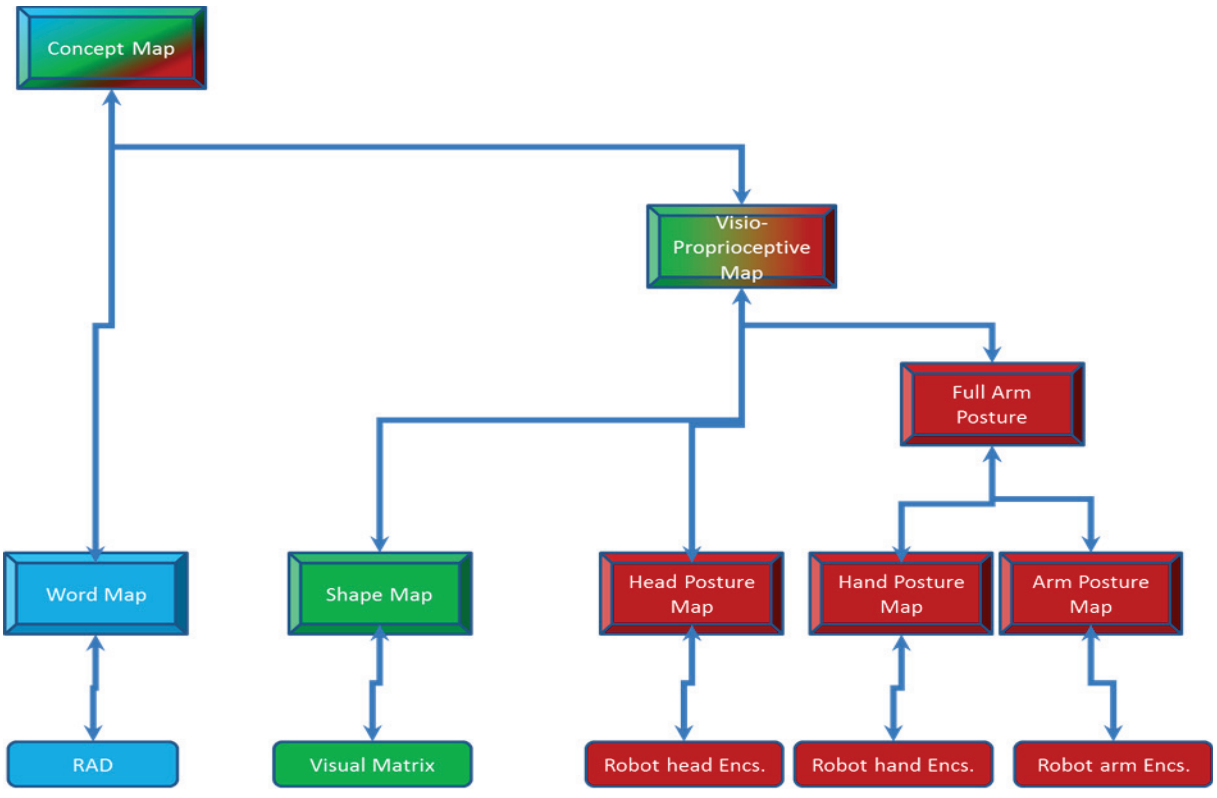


Figure 18: A potential hierarchical organization of MMCM. Using this kind of arrangement high level amodal concepts could be coded in a cortex-like fashion.

Since a map of neuron can be used as a modality input for another map, the question of how to interpret a given map activation can be asked. It can be described as multimodal percept, compressing vectors of low level features, but I prefer to refer to it as a concept. When one is asked about giving a definition of a dog, he says that a dog is its image, its smell, semantic information (number of legs, color, can bark, etc.), everything that all dog possess and which can be coded as activation of modalities. Of course those modalities will always be congruent when observing a dog, therefore strengthening the concept of dog in the subject. There is no doubt that in some place in the cortex, a population of neurons is coding for the dog concept, but I do not say that those are the "dog neurons". Those neurons are nothing by themselves, but they link together a rich hierarchy of amodal neurons in order to be finally expressed as modal neurons. When I want to imagine a dog, my brain may be activating this initial population and the feedback cascade could be the origin of this dog mental image that appears if I wish to see it. Although MMCM can be clearly qualified as a "connectionist" model, it is solving the symbol grounding problem exactly as described in (Harnad 1990) by using an hybrid symbolic/non symbolic representation. Modalities, when linked to the sensors, are clearly providing non symbolic information to the maps, however after learning and convergence, a given map activation can clearly be understood as the neural code of a symbol grounding the associated modality activities. Moreover, this neural code itself can be used as non-symbolic information feeding a higher level map, while still being interpretable as a pure symbol. Indeed MMCM provide a tool to symbolize the embodied experience of the robot and enable higher cognitive functions to work on those symbols. Although it is in theory possible to implement also those functions using an enormous hierarchy of neural maps, once the world symbolization has been achieved, cognition can be achieved using classical software engineering methods as we will see in the next chapters. Finally the MMCM can be used as a model of multimodal integration in the cortex and allows producing and verifying several hypotheses about modalities interaction and their behavioral consequences. It can also provide a very good model of the synesthesia phenomenon (Cytowic 2002; Nunn, Gregory et al. 2002) which it will explain in two ways: first synesthesia could be the result of a wrong connectivity pattern (two cortical areas that are not supposed to be connected occurring in being linked). Another option, assuming the enormous rate of connectivity in the brain, is that an abnormally high correlation between statistically unrelated modalities is experienced early on during development, therefore

allowing the child cortex to encode relations between modalities that are unrelated. Although the first hypothesis might be tested using emerging imagery techniques like DTI, the second one is likely to be impossible to test. Although it is beyond the scope of this thesis, it could be exciting to design a robot with synesthetic capabilities, and to see if it is possible to “cure” it by imposing an “unassociative training”. Due to plasticity of the model, such a treatment should be possible; therefore it could be a good investigation to carry on in the human case.

Chapter II

Symbolic Action Definition, from Primitives to meaning

Introduction

As mentioned above, one way to benchmark the intelligence of an agent is to examine and quantify the characteristics of its interaction with the world, thus, the central importance of action. Before the more scholarly treatment below, we begin with some intuitions. Every action has consequences, an effect on the surrounding world, and a principal activity of living beings consists in choosing which action to perform at any given time. The pool of possible actions is defined by the agent himself (a fish will never be able to grasp an apple) and by the state of the world (one will never be able to grasp an apple while standing in front of a banana tree). Among all the possible actions, defining the right one is a matter of goal, of which world state we would like to approach. We are able to make this choice because we know what we can possibly do and which effect it may have on the world. Of course, whenever we act we don't always check all our possibilities, nor what could be the consequences of them, our behavior is triggered by the world state and our inner universe. Rather than how one will choose the sequence of actions that should lead to the goal, this chapter will focus on a single action. We will try to grasp what is the concept of action and derive an acceptable model for it. As stated before, an action possesses preconditions or requirements and consequences, but those are not sufficient. If I decide to grasp an apple while you are observing me, my brain will send highly complex commands to my muscles in order to execute my will. At the same time, your cortex will treat the sensory information received and you will notice that I just grasped an apple. You may even notify your neighbor who was not looking by telling "*Hey! Stéphane grasped an apple!*". These are basic functions of an action: we can execute it, perceive it or describe it. We want the robot to manipulate actions as humans do. That is why it should possess these three abilities. So logically this chapter will be articulated around these major axes:

1) Perceptual: when an agent performs an action it can be perceived and recognized by surrounding agents.
2) Motor: actions are the way for agents to interact with the world; their representations need to embed which command has to be sent to the effectors (muscles, motors…) in order to produce the action.
3) Descriptive: for human beings, actions can be described using spoken language. This ability requires linking the data structure recognized by (1) and produced by

(2) with symbolic representations (words) for both the action and the potential arguments.

For each of those aspects of action, the literature is reviewed to outline how they are achieved in the human in term of cortical connectivity and how the child learns them. Indeed, we don't simply want the robot to be able to recognize, execute or describe a specific action: we want it also to be able to learn how to do that for every possible action in its sensory-motor and perceptual space. On one side, neurophysiology gives us directions about the flow of information within the brain and it can be used to extract which concepts are shared among different functionalities, therefore it is clearly a useful guide to the software conception at the structural level. On the other hand the psychological experiments carried out with children inform us with important information about which behaviors are making use of these structures to perform an efficient learning through interactions with other agents. My goal in this chapter is to give a symbolic definition of action, but also to show how it can be used in order to allow the robot to populate a knowledge base through its interaction with human beings.

Action Definition: Perceptive Level

To perceive and furthermore recognize an action, an agent needs to interpret the stream of his perceptions and match what is being perceived with some symbolic representation it has previously stored. It is important to notice that the term perception is not linked with any specific modality or sensor. Much of the work on action perception is done on vision and we will use this specific modality in our explanations for ease; however an action can be recognized using other type of information (e.g. consider hearing someone climbing wooden stairs).

Anatomical

Anatomical neural networks involved in action perception have been studied under a variety of different conditions. Decety and Grezes (Decety, Grezes et al. 1997) have shown that the content of action (determining whether it is meaningful or not) as well as the observation strategy (do we watch the action in order to recognize or to imitate) do not involve the same cortical structures. In this specific study the actions used were pantomimes, which are sequences of motions performed by an agent, however objects were not present and only suggested by the motion pattern. In one condition the pattern was that of an object directed action, so called a meaningful action. In the other condition meaningless actions are arbitrary patterns of motion. In the case of meaningful actions they observed "in the left hemisphere a ventral visual pathway which includes inferotemporal areas, part of the hippocampus and terminates in the ventral part of the prefrontal motor cortex"; on the other hand the meaningless actions (sequences of motion) produced activation in the right hemisphere along a "dorsal pathway including occipitoparietal areas and is connected with premotor cortex cuneus and the inferior temporal gyrus. Thus, the ventral stream also contributes during the observation of meaningless action." In this section of my thesis I will mainly put the emphasis on so called "meaningful actions", because this kind of actions embeds semantic information. Indeed, Decety et al. reported that observation of meaningful actions (on both recognition and imitation purposes) involved the temporal area 21 (semantic object processing). Area 45 of the left inferior frontal gyrus is also involved, it is known to be used in tool recognition (Perani, Cappa et al. 1995) and to represent grasping movement (it is the human analogous of the ventral area 6 of the monkey) and more generally hand related movements (Grafton, Arbib et al. 1996).

Meaningful actions have a semantic value and they often involve objects. In order to interpret a sequence of motion resulting in object manipulation, it is required to have a way to access information about objects. This is why the action recognition network involves areas that are known to process information about objects identity and properties: it is required to interpret an action in a semantic way. In the implementation part we will see that the action recognition system of the robot is based on perceptual events about objects. The fact that some of the above studies rely on pure motion of human limbs, without involving objects, is not in contradiction with our approach. Those studies involve "imagined objects" and therefore they recorded activation in object related areas of the cortex. Moreover, for within our system an agent limb (the human hand for example) is just a specific object. Indeed, the robot perceives as object every "world entity". The work of Decety and Grezes give us important insights about how the brain is segregating meaningful from meaningless actions during observation. At this stage we can already extract parts of the network responsible for goal attribution process and therefore the classification of actions as being goal directed or not. It is important to note that goal directedness can emerge from pantomime motion, showing that the goal attribution is based on interpretation of physical trajectories among the entities present.

Although in this thesis action recognition modeling will be based mostly on objects' physical relations (Faillenot, Toni et al. 1997; Shmuelof and Zohary 2005), it is likely that a great part of understanding others action in animals is achieved using a mirror system mechanism. This is developed further in an Annex, however in the mirror system, the cortical body representation is similarly activated when both producing and observing an action (Rizzolatti and Arbib 1998; Decety, Chaminade et al. 2002; Rizzolatti and Craighero 2004). Indeed there is a growing amount of evidence arguing that one understands others behavior by mapping their actions and body schemas on one's own, simulating what others are doing using ones self-representation, a kind of understanding through physical empathy (Grezes and Decety 2001; Calvo-Merino, Glaser et al. 2005; Calvo-Merino, Grèzes et al. 2006).

Developmental Psychology

The action recognition skill is acquired very early in the child development. Although we cannot really speak about "recognition", it has been shown by Woodward (Csibra, Gergely et al. 1999; Woodward 1999; Király, Jovanovic et al. 2003) that infants are able to detect

actions and even classify them regarding their goal-directedness starting from the age of 6 months. Mandler (Mandler 1992) suggested that the infant begins to construct meaning from the scene based on the extraction of perceptual primitives. From simple representations such as contact, support and attachment (Talmy 1988) the infant could construct progressively more elaborate representations of visuo-spatial meaning. In this context, the physical event "collision" can be derived from the perceptual primitive "contact". Kotovsky & Baillargeon (Kotovsky and Baillargeon 1998) observed that at 6 months, infants demonstrate sensitivity to the parameters of objects involved in a collision, and the resulting effect on the collision, suggesting indeed that infants can represent contact as an event predicate involving agent and patient arguments. (Allen 1984; Mandler 1992; Allen and Ferguson 1994; Siskind 1998). Indeed, even the basic definition of what an object is will rely on those physical attributes and relations that are inherent to all physical entities. For example, isolating an object within a scene, or a small independent part in a bigger object is based on physical bounds and co-motion, and those are perceived and used during early infancy (Kellman, Spelke et al. 1986; Spelke 1990; Spelke, Vishton et al. 1995). Indeed those physical properties are intrinsically linked to the notion of perceptual primitives: it is not clear if we know there is an object A and we can observe that it is moving, or if we know that this shape is an object because it is made of points that are all moving in a coherent and natural way. Spelke isolated a reduced set of (Spelke, Vishton et al. 1995) physical behaviors that characterize an object for infants and those are mainly related to the perceptual primitives we are interested in, as if these early perception primitives was not a consequence, but a cause of object perception. That is to say that the ability to perceive physical relations between objects like contact, occlusion or co-motion seem to be present very early in infants cognition and is likely to serve as a building block for higher level perceptual constructs like an action. Indeed, a framework integrating the perception to the action based on the fact that an action can be described as a succession of events has been designed by Hommel et al. (Hommel, Müsseler et al. 2001). According to the Theory of Event Coding an action is made both of event codes and action codes which are more or less similar to our definitions of perceptual and motor primitives.

Robotic Implementation

While many action recognition systems are based on kinematic motion pattern (Gavrila 1999; Moeslund and Granum 2001; Schuldt, Laptev et al. 2004), only a few take as input objects perceptual events. Siskind (Siskind 1998; Siskind 2001) demonstrated that force dynamic primitives of contact, support and attachment can be extracted from video sequences and used to recognize events including pick-up, put-down, and stack based on their characterization in an event logic. Related results have been achieved by Steels and Baillie (Steels and Baillie 2003). The use of these intermediate representations renders the systems robust to variability in motion and view parameters. Based on previous work, we(Dominey and Boucher 2005; Dominey and Boucher 2005) have used a related approach to categorize movements including touch, push, give, take and take-from in the context of linking these action representations to language (Lallée, Madden et al. 2010) (attached as Appendix 3). This section provides a deeper explanation about these perceptual events: how can they be detected, how can they be characterized and what can we use them for.

We call perceptual primitives those intermediate representations; they are events that produce a salient physical change in the world state which mean that some properties of one or more objects are changed enough and in a sufficient fast way to attract attention and to serve as a base for encoding meaningful segments of observation. They can be computed given the evolution of the world state through time. In order to understand the notion of world state for the system, we will explain how objects are represented within our robotic cognitive architecture. A data structure called Egosphere contains the status of every object perceived by the robot. At the symbolic level, objects are structures containing a name and a list of properties (position, orientation, isVisible, isMoving, isContainedBy, isTouching, etc.). A perceptual primitive can therefore be described as a salient change within the spatial properties of an object. Indeed, these events can be seen as the derivative of world state over time: changes that occur among the status of objects. At the physical properties level, an object can appear or fade (visibility), it can start to move or stop (motion) and a physical contact can be established or broken between two objects (collision). Subsequently, the data structure modeling a primitive should provide us information about which object properties are altered and in which way. While the full Action data structure is quite complex (See for reference Figure 25 on page 79), the perceptual side is simple: a

perceptual primitive is only a list of object properties modifications. A property modification is described by the name of the object, the property and its new value (e.g: *(name toy (isVisible true))* will describe a toy appearing). Using a list instead of a single property modification allows us to characterize all the physical primitives (which are basically a single property modification) as well as more evolved consequences of action. Indeed I will now describe the process of recognition and demonstrate that an action should be described as being the sum of all the perceptual primitives triggered by its execution.

Figure 19: Visual/name binding (A, B) and use of those bindings to proceed to action learning and recognition (C) on the BERT2 platform in Bristol.

Let's setup a simple situation as the one presented Figure 19. Subject is facing the robot and a set of objects are lying on the table between them. When the subject grasps the toy (Figure 19.C) the Egosphere dataflow along the action is the one described on the Figure 20.

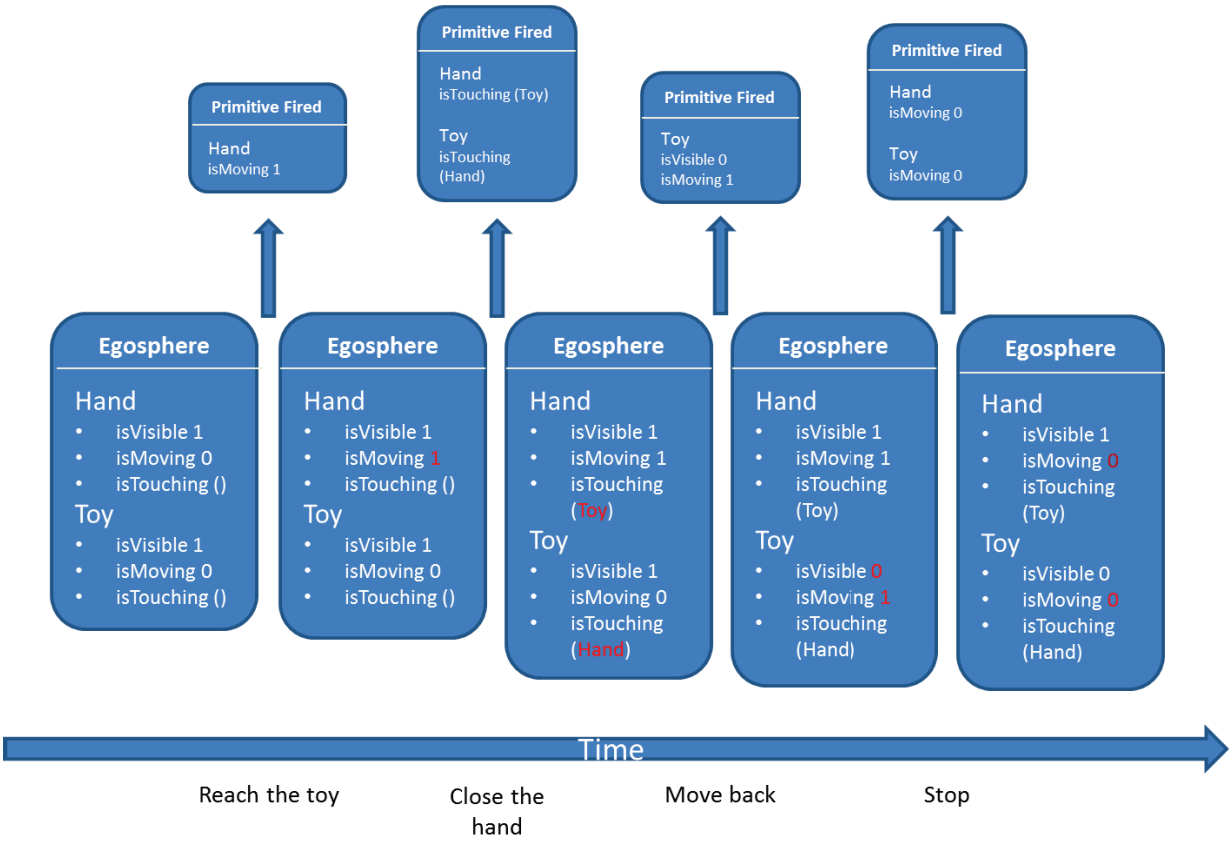


Figure 20 : Evolution of the Egosphere status (perceptions of the robot) over the grasp(toy) action. Several perceptual primitives are detected.

It shows that 4 perceptual events are triggered, in that order: motion of the hand, contact between hand and toy, co-motion of hand and toy, stop of motion.

The role of the Action Recognizer is to catch those events and interpret them. The first problem appearing is the segmentation of the continuous events stream. We need to parse the stream coming out of Primitive Recognizer in order to segment a sub-sequence of events which could potentially describe an action. We assume that only one action will occur at a given time, and that two actions will be separated by a small delay. Typically the beginning of a segment is an event being received after 3s without events, and the end occurs when no event is received for 3s. This timing interval is a parameter of the system, and the value of 3s has been determined experimentally. This segmentation is quite artificial in our implementation since we just detect “paquets” of events which occur in the same time interval; however a statistical segmentation could also be used and would probably be more robust. This segmentation problem has been investigated by many researchers and the

statistical analysis solution seems promising and biologically plausible (Rui and Anandan 2000; Saylor, Baldwin et al. 2007; Baldwin, Andersson et al. 2008; Shi, Wang et al. 2008).

Once a segment (a sequence of events) has been detected, the recognition process can be applied: the goal is to match a sum of perceptual changes with the name of an action and its arguments. For manipulation ease I defined a few mathematical / programming operators that can be applied on the Perceptual Primitives.

Two perceptual primitives can be added, and the result of this addition will be a third primitive containing the sum of their Properties Changes (e.g *hand(isMoving 1) + toy (isVisible 0) = (hand (isMoving 1) toy (isVisible 0)))*. The addition of primitives is a base for any teleological reasoning capability as we will see later on in Chapter 3: understanding the overall effect of a sequence of primitives (i.e the global change of world state induced by this sequence of primitives) is a matter of adding all those primitives together. The addition process is fairly simple and a few examples are given in Table 1.

A	B	A+B
Toy (isVisible 1)	Toy (isMoving 1)	Toy ((isVisible 1) (isMoving 1))
Toy (isVisible 1)	Box (isVisible 0)	Toy (isVisible 1) Box(isVisible 0)
Toy (isVisible 1)	Toy (isVisible 0)	Nothing
Toy (isVisible 1)	Toy (isVisible 0) Box(isVisible 0)	Box (isVisible 0)

Table 1 Examples of the + operator on Perceptual Primitives

It is also possible to test the equality of two primitives, by checking if all their property changes matches. However the equality test is not sufficient since we are working with relative arguments: we want another operation that will classify *(toy (isMoving 1))* and *(box (isMoving 1))* as being equal under the condition (*original argument toy = box*). Therefore we discriminate the absolute equality (strict equality for both properties variations and arguments) from the relative equality (properties variations match, but arguments are changed). These equality operators work for both a single primitive and a sequence of primitive (i.e a perceptual action) as it is described in Table 2.

	Stored Pattern Sequence A	Observed Sequence B	Observed Sequence C
	Box isMoving 1	Hat isMoving 1	Hat isMoving 1
	Toy isVisible 0	Head isVisible 0	Head isVisible 0
	Box isMoving 0	Hat isMoving 0	Head isMoving 0
Recognition	Cover(Box,Toy)	Cover(Hat,Head)	None

Table 2 : Using the relative equality, A == B but A != C. The arguments are set by taking each argument of the current element of the sequence in the temporal order and matching it with the original argument (Box →Hat, Toy→Head). Using absolute equality give us A != B and A != C

Those two operations were proven to be a very handy tool for the action recognition process. Indeed the recognition process can be asked to perform two different tasks: recognize any action, or wait for a specific action to be recognized. Figure 20 shows that the perceptual aspect of an action can be reduced to a list (sequence) of Perceptual Primitives. The action recognition process stores a database of all known actions using the arguments used when they have been teach for the first time. Whenever a segment of primitives is isolated, it is tested against all known actions using the relative equality, therefore providing an argument independent recognition of action. Testing if the detected action is the one we were waiting for is just a matter of using the absolute equality operator between them. A detailed explanation of the recognition algorithm will be presented in the experiment description about imitation.

Action Definition: Motor Level

How does an animal proceed from the desire of an action (walking, grasping, etc.) to a motor command that will make the muscles follow the right pattern to produce right motion? This question has been, and is still being investigated. It is broadly accepted that high level commands are decomposed hierarchically into low level controllers that we will call Motor Primitives. We will go though neuro-anatomical and developmental literature to identify evidence for those primitives, then we will explain how those results are implemented into our robotic architecture and what the advantages of such an approach are.

Anatomical

Many studies demonstrated that animals use a hierarchical decomposition for achieving desired motion. This decomposition occurs at multiple scales, ranging from the effective motor activation to generation of higher level commands which I would be tempted to call planning. A remarkable review on motor primitives has been provided by Flash & Hochner (Flash and Hochner 2005). They describe the different levels of compositionality in movement generation both in vertebrates and invertebrates. Their definition of motor primitives is clear and general: "Motor or movement primitives refer loosely to building blocks at different levels of the motor hierarchy. Motor primitives might be equivalent to 'motor schemas' (Arbib 1998), 'prototypes'(Jeannerod, Arbib et al. 1995), or 'control modules'(Schaal, Ijspeert et al. 2003)."

At the lowest level, Mussa-Ivaldi has shown evidence that the frog's spinal cord stores a pool of motor primitives (Mussa-Ivaldi, Giszter et al. 1994). Mixtures of those primitives are called in linear combinations by the central nervous system in order to execute more complex behaviors. In the human, the same process of primitive encoding may occur in Purkinje cells in the cerebellum (Mussa-Ivaldi and Bizzi 2000). At the cortical level, electric stimulation of the premotor and motor cortex in the monkey resulted in arm movements which were similar to the standard behavior of the animal (Flash and Hochner 2005), suggesting that those parts of the cortex may be maps encoding for different combinations of standardized primitives.

In our case, the Motor Primitive concept is situated at a higher level that has not really been investigated in biological beings. We considered as primitives chunks of actions, like "grasp"

or “release”, which are often referred as complete actions in the literature. However, the main idea is to use compositionality, which is to compose with building blocks in order to achieve a more complex behavior. What Flash & Hochner (Flash and Hochner 2005) have shown is that this principle is present at many level in biological beings, both on motor, language and sensory sides ; we just extend this idea one level higher. One could argue the fact that at this level, what we call primitive is already a complex sensory-motor process involving both motor control and perceptual feedback; indeed it is the case but the compositionality principle is still valid: those sensory-motor processes can be seen as atomic functions which can be sequenced to produce complex behaviors. Maps of those high level primitives’ symbols can be found in the cortex; indeed in the part Action Definition: Descriptive Level we will see that this symbolic definitions are located in language areas which are known to be involved in the composition process. A feedback mechanism similar to the one of the MMCM (see first chapter) could be activating lower level motor primitives in a sequence.

Devlopmental Psychology

The motor development may be a principle source of pride for parents during the first year of their child. At those times a child learns to fix his balance in order to sit properly, to control each of its limbs as a semi-independent effector and to coordinate all of them in order to grasp an object or to crawl on the floor. Since motor development is one of the oldest fields in developmental psychology, the amount of literature on the topic is vast and not all authors are in agreement. Thelen synthesized more than 50 years of literature in (Thelen 1995) from which I would like to emphasis the part on composing evolved behavior on the top of simple motor primitives. We will see all along this thesis that compositionality is a generic computational mechanism used in many places of the human cognition. Based on small building blocks, we can produce higher level structures which can be used as building blocks in a recursive fashion. Although this principle is referred to differently by different researchers, it was first pointed at by Bernstein (Bernstein 1927) by asking a question about redundancy in ways to achieve a motion pattern and how does the brain handle it. Hypotheses have been made and the one which seemed the most interesting and congruent with our framework is the work of Sporns and Edelman (Sporns and Edelman 1993) which describes how this problem is likely to be solved by using a “*repertoire of*

motion patterns". For easy referencing we present in Figure 21 a quite self-explanatory picture coming from (Bernstein 1967): it describes primitives as pattern of movements (making a circle, a letter, a segment…) that are independent from the collection of muscles used to produce them. They are high level commands to motor controllers and can be executed with various parameters including the spatial position, the speed, etc. Moreover we can see on this example (drawing a star) an expression of the compositionality principle: the motor primitive used could be "draw a segment", this primitive repeated five times with various spatial parameters produces a star. In addition, the "draw a star" action is another chunk of motor command that can be called and that wraps those lower level commands.

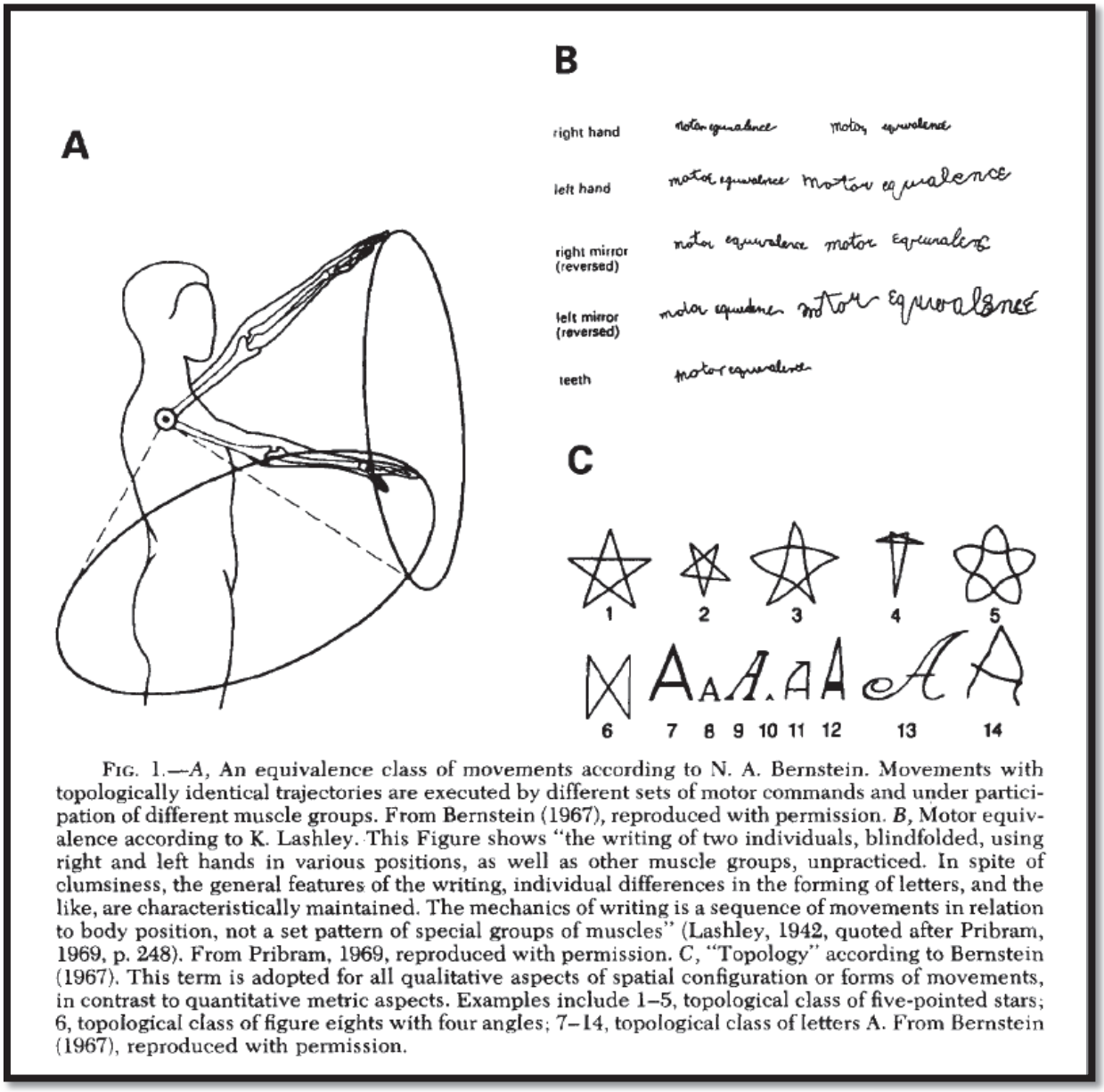


FIG. 1.—*A*, An equivalence class of movements according to N. A. Bernstein. Movements with topologically identical trajectories are executed by different sets of motor commands and under participation of different muscle groups. From Bernstein (1967), reproduced with permission. *B*, Motor equivalence according to K. Lashley. This Figure shows "the writing of two individuals, blindfolded, using right and left hands in various positions, as well as other muscle groups, unpracticed. In spite of clumsiness, the general features of the writing, individual differences in the forming of letters, and the like, are characteristically maintained. The mechanics of writing is a sequence of movements in relation to body position, not a set pattern of special groups of muscles" (Lashley, 1942, quoted after Pribram, 1969, p. 248). From Pribram, 1969, reproduced with permission. *C*, "Topology" according to Bernstein (1967). This term is adopted for all qualitative aspects of spatial configuration or forms of movements, in contrast to quantitative metric aspects. Examples include 1–5, topological class of five-pointed stars; 6, topological class of figure eights with four angles; 7–14, topological class of letters A. From Bernstein (1967), reproduced with permission.

Figure 21: Figure taken from (Bernstein 1967; Sporns and Edelman 1993). Represent well the concept of motor primitives used to compose higher level action (writing a word, drawing a shape).

The fact that primitives are not a frozen sequence of postures (for example a simple sequence of joints in a robot) but are adapted to a spatial target and more generally to the environmental context is a quite important element regarding the perception of the produced action. Studies of human infants (Csibra, Gergely et al. 1999; Király, Jovanovic et al. 2003) demonstrated their ability to attribute goal directedness for novel actions early assuming two conditions: first the action has to produce a salient effect on the world state. The second condition is that the agent is able to achieve the same state change in different ways (such as avoiding an obstacle instead of using a straight trajectory), in other words the

action is demonstrated to possess equifinal variations. Our implementation of action, both in the context of perception ((Lallée, Lemaignan et al. 2010), Appendix 1) and execution is based on actions seen as state changes. One of the strong implications of this is the equifinality of action. That is, the same action "put the box on the toy" may be realized in a variety of ways (with one hand, or the other) but with the equivalent final outcome, one of the key characteristics that allow action to be considered goal directed. If the robot is able to demonstrate equifinal means of achieving his actions, then humans may be more likely to attribute a goal to them. This assumption has been shown to be true in infants (Király, Jovanovic et al. 2003; Kamewari, Kato et al. 2005) and would need to be tested on adults, however assuming the fact that the human teleological system seems to be built on those core capabilities it is likely that a benefactor effect could be found also on adults.

Robotic Implementation

Motor primitives rely on the idea that complex motor tasks may be achieved by the combination of simple parameterized controllers we call primitives. Using hierarchies of primitives for control in robotics is becoming a widely used method (Firby 1992; Williamson 1996; Mataric, Williamson et al. 1998; Mussa-Ivaldi and Bizzi 2000; Morrow and Khosla 2002; Thomas, Finkemeyer et al. 2003; Paine and Tani 2004; Sentis and Khatib 2005). In our approach, what we call a Motor Primitive is already a fairly high level procedure, the first level of symbolic actions. Most of the experiments conducted within this thesis implied a robot and a human interacting together in a shared work space over a table. Focusing on this limited interaction, we were able to define a pool of motor primitives that were enough to compose evolved actions and cooperative games. The identified set was:

- Grasp (object)
- Release (location)
- Touch (object)
- Look-At (object)

Of course this pool is not complete enough to cope with real world human robot interaction, for example with a mobile platform the primitive Move-To(location) would be essential. For historical reasons primitives composing this set have as special status, they have been defined as functions of an abstract class, so that each of several different robots

can inherit it and have implemented its own controllers. This provides platform abstraction on the motor side of the action definition. However it is a handy tool in interaction with a robot to be able to define new primitives so we decided to implement the ability to teach new primitives to the robot within the interaction framework. In the end a motor primitive is what makes the robot motors to move: a sequence of either joint angles or velocities over time. So if I want to teach the robot how to "wave to Peter" I can either wave myself to Peter and have the robot to imitate me at the joint level, or I can physically take its arm and move it the way I want. The first way (imitation) is likely to be the most commonly used for humans to learn primitives from others (Meltzoff and Moore 1989), although many primitives are probably discovered by lonely interaction with the world on the basis of an existing "innate pool" of very basic motor controllers. The second way (kinesthetic teaching) can appear a bit artificial and not inspired by human behaviors, however some reeducation therapies use it. In both cases the goal is to have the limbs of the subject to move so it can perceive it and record the motion pattern. Basically when the robot learns a primitive, it is placed in a "recorder mode" which will record joint angles or velocities of the robot limbs at a given rate. As shown in Figure 22, the primitive motion is then demonstrated either using imitation (the demonstrator skeleton is tracked using Kinect for example and then mapped to the robot one) or by physical interaction (the robot body is set to compliant mode, and the desired motion is achieved by moving it manually). The recorder keeps track of the demonstrated trajectory and links it to the primitive name, making it straightforward to playback later.

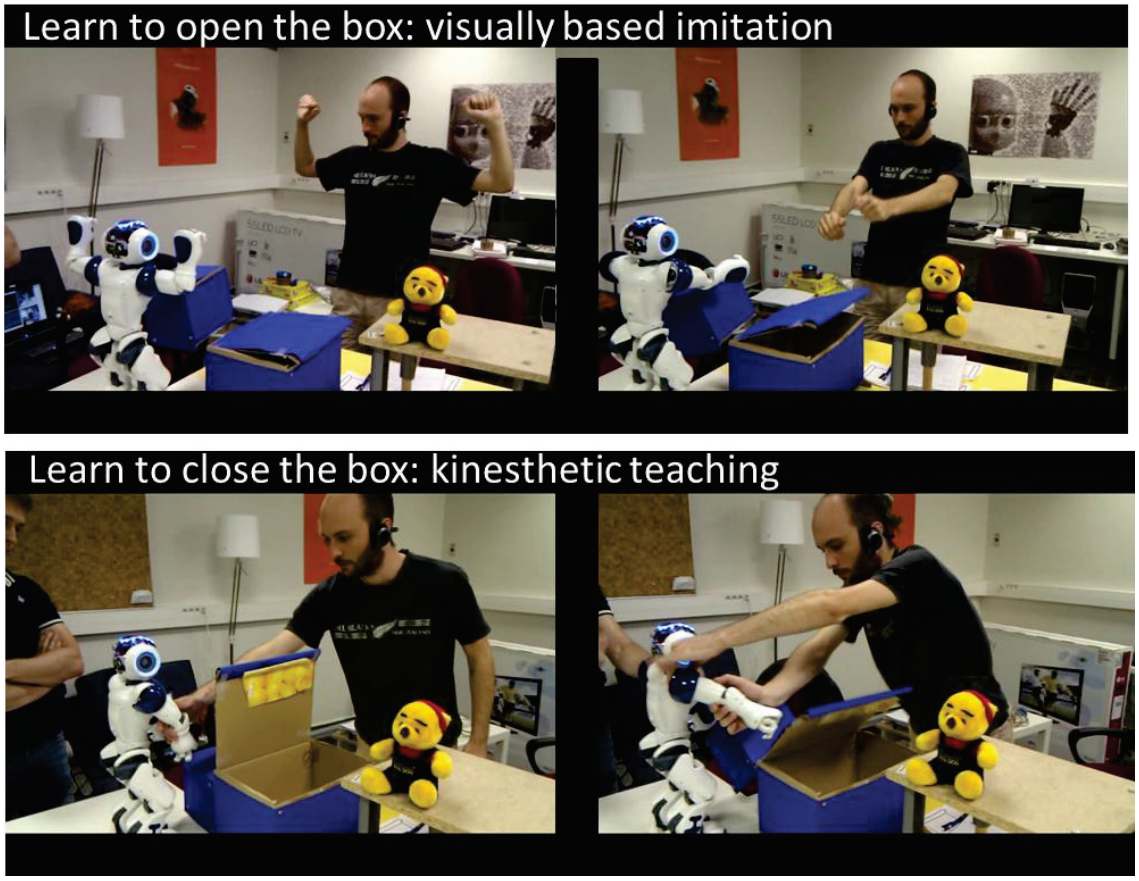


Figure 22: Nao learning to open and close the box using two different modalities (visually based imitation and compliant kinesthetic teaching)

However, while recording an animation (wave, dance, etc.) relies on this simple process, teaching a primitive that takes an argument is a bit more difficult. Assume that we want to teach the robot how to grasp. We can say to him "I will teach you how to grasp the toy", then the system will prepare the primitive *grasp(toy)* to be learnt. Since the primitive has an argument, the limbs trajectory is not absolute: it is relative to the object position. That is the reason why in the case of primitives taking an argument, the pattern recorded is composed of displacements between the position of the robot's end effector and the position of the argument (see Figure 23). However, this solution has two major issues: it requires that the robot has a cartesian controller[7] implemented (which is the case for most robots today, at least the humanoids one) and it cannot cope with primitives that could use more than one argument. Anyway, this last remark is not really a problem, indeed we argue that using

[7] In robotics, this is a controller that can calculate the robot joint angle trajectory necessary to reach a target point in cartesian (working) space from the current position.

multiple arguments is beyond the complexity scope of primitives; such commands should be regarded as actions and built on top of motor primitives.

Since motor primitives will be the basis for action, they also embed some « built in » reasoning knowledge. At the physical level, motor primitives are constrained; they require a certain state of the world in order to be executed. For example, I cannot grasp a toy if there is no toy or if it is stuck under another object. For this reason the Motor Primitive data structure also embeds a list of (pre) conditions. Those conditions are what the robot needs to check in the world state before a motor primitive is executed. While the natural way to learn those conditions would be again trial and error and statistcal learning, we decided to speed up this process by hard coding some basic conditions into the pool of primitive defined above. Most of them possess the conditions (argument isVisible==1) and (argument isContained == 0). However, hard coding is possible only if you designed the system: the final user should be able, after having taught a primitive, to specify to the robot under which condition it can be executed. It can be done using speech by telling sentences of the form:

*If you want to **primitiveName** the **argument**, then the **argument** should be **condition***

*If you want to **primitiveName** the **argument**, then the **argument** should not be **condition***

As we will see later on, those conditions on primitives will serve as a basis for determining if an action is possible or not, therefore opening the door to reasoning. Indeed an action will always be achievable under the sum of the conditions of the primitives it is composed of and possibly its own conditions.

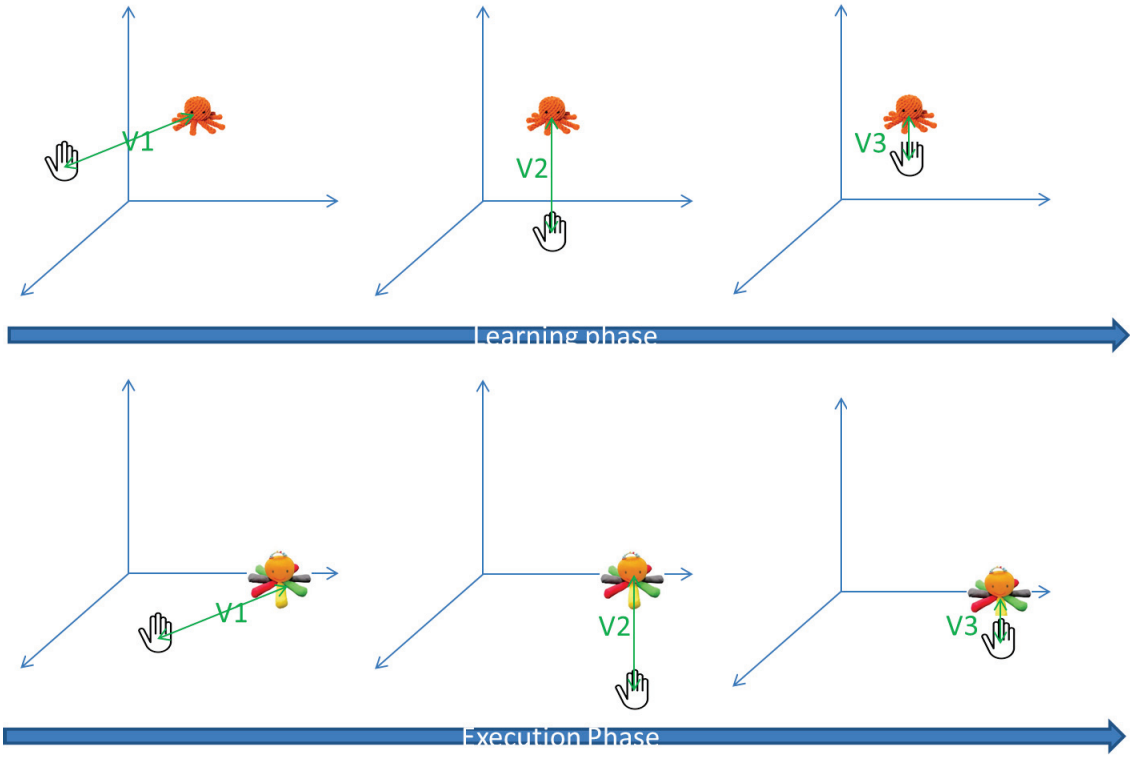


Figure 23 : Teaching of argument dependent primitives. At each time step the displacement vector between the target and the hand effector is computed and recorded. This same vector is replayed later on, relatively to the position of the new object.

Action Definition: Descriptive Level

Actions are sequences of motion that have an effect on the world and that can be either executed or perceived by an agent. The auditory aspect is also deeply linked to the action representation (Kohler, Keysers et al. 2002). In the case of agents endowed with speech (humans, robots), an action can also be linked with a spoken representation so that the agent is able to describe or hear what is going on. Humans are able to describe actions that are part of their repertoire using the label they associated to this action; they are also able to describe an action they do not know by splitting it into pieces they are able to describe or by commenting on the perceptual effects produced. In this section we will first study the neural correlates of this ability, and then how we implemented it in the robot architecture.

Anatomical

The vast action recognition and production cortical network involves Broca's area, which is known to be dedicated to language processing. Arbib exposed a theory arguing that language might have evolved from gesture imitation instead of pure vocalization (Arbib 2005) ; the motor theory of language fits perfectly with this approach. Assuming that animals from the same species have a tendency to imitate each other, due to the mirror system, imitation of the speech production system motor activity would facilitate the vocabulary grounding. Moreover, the hierarchical organization of motor primitives into more complex actions and the grammar produced and used in language are both making use of the compositionality principle. A specifically important cortical area involved in the binding between language and action is the left inferior frontal girus (LIFG) (Hagoort and Van Berkum 2007; Willems, Özyürek et al. 2007; Hagoort, Baggio et al. 2009). As we will see later, this area may be dealing with the compositionality principle in general, for speech, primitives, shared plans, etc.

A number of studies have started to establish the link between action and language. Tettamanti et al. (Tettamanti, Buccino et al. 2005) demonstrate how action verbs activate the area of the premotor cortex associated with exectution of these actions. Similar resutls have been observed by Pulvermuller (Pulvermüller 1999; Pulvermüller 2005). We can consider that these are consistent with an em bodied view of cognition (Zwaan and Madden 2005; Barsalou 2008). This view would account to say that words are cues that re-activate sensorimotor representations. Therefore the co-activation of the motor primitive "grasp"

and the perceptual primitive “apple” given by speech may lead to the physical act of grasping an apple if there is one available. Again the anatomy of an action description cannot be viewed as a single area: at best it is a convergence zone where a specific pattern of activation links together the physical reprepresentation of an action parametrized by the other sensory traces of the arguments involved. Indeed an action is probably stored within a quite large set of cortical areas where different subregions codes for different combinations of arguments. This is purely speculative, however it would be consistent with the various findings about convergence zones and hierachical traces storage presented in Chapter 1 .

Developmental

I’ve recently been watching a three month old baby and his parents. Apart from storytelling, their spoken “interaction” was mainly the caregiver describing what he was doing or what the child was doing in simple sentences of the type *subject verb [object].* Adults rarely describe their physical actions when interacting together, unless they want to teach (e.g. how to fix a robot shoulder). There is an assumption that our actions are self-explanatory. In the case of a baby/adult interaction it is quite amazing to see how parents often verbally describe their actions or the actions of the child, with sentences that are used in return by the child (Gleitman 1990; Mintz 2003). When a baby grasps an object, he may learn to associate visual, proprioceptive, tactile modalities consequences allowing him to use the grasp action in appropriate situations, however it is hard to imagine how he may learn the spoken description of this act unless someone tells him. Therefore the adult tendency to describe low level actions in the context of interaction with a baby could be facilitating the acquisition of perceptuo-motor acts linked and labeled as a meaningful action, indeed this joint attention process can be valuable for both action and object labeling (Tomasello and Farrar 1986). While such an exhaustive description of the world state is required for vocabulary acquisition, such useless description are kept silent at later age, allowing the children to learn what doesn’t need to be told (Aukrust 1996).

On the motor side, I also noticed that when the child was doing some specific posture, like joining hands in a “clap”-like movement, the caregiver was imitating him. In this case he didn’t say anything to describe this action, but if those two behaviors are combined, it is a way to relate an action spoken description and the corresponding motor act. Children imitate adults from a very early age (12 days)(Meltzoff and Moore 1977), however I was not

able to find in the literature studies about adults imitating babies. Maybe is it too evident to be investigated, however the adult tendency to do "low level action naming" and to imitate low level actions of their baby seems an optimal behavior to ground action and language. Indeed while grounding of nouns is natural and just a link between perceptions (visual and sound for example), verbs are usually though as being more difficult to learn (Gentner 2006) because they refer to concepts that are more abstract or that are continuous sequences of perceptions. However, it appears that the same mechanism of co-occurrence of spoken description and the object of interest (an object, an action, a more abstract concept) in various contexts provides a unified way to teach children vocabulary and language (Maguire, Hirsh-Pasek et al. 2006). Moreover, this is a valuable mechanism to model the mapping between the self and the other: the way to teach artificial systems by imitation is mainly by having the system to perceive the caregiver and try to reproduce what it sees; a design where the system produces a behavior that is afterward imitated (in an improved form) by the caregiver could be a nice alternative (refer to Annex 1: A Theory of Mirror Development for a motor mapping experiment based on this principle).

Implementation

While the previous parts of this chapter described how the motor and perceptual parts of an action were embed into a single data structure, it still lacks of information about the spoken description of an act. In this part we give details about the implementation of the ability to generate sentences that describe an action. Those sentences can afterward be used either for expression (text to speech) or understanding (speech recognition).

The Action data structure contains a label for easy access and dictionary-like storage of the set of known actions, this label is a verb describing the action. While storing a verb to describe an action can be enough to use grammars in order to produce spoken sentences, this approach turned out to be problematic. A grammar can be used to recognize or produce sentences in a generic way; however this genericity has a tendency to produce syntactic errors. Verbs involve prepositions (to pull from the box, to cover with the box), use different number of arguments (to wave at someone, to put the toy in the box) and therefore won't use exactly the same generative grammars. In order to produce correct sentences, they need to be split into classes (verbs with 0,1,2 arguments, followed by certain prepositions, etc.) after analysis of the verb type. This corresponds to the link between syntax and semantics

that is captured in certain lexical function grammars (reference). However, in the case of action learning the robot can benefit from heuristic that allows it to produce the right sentence. When the human teaches an action to the robot, he is first asked the name of this action, through open dictation or spelling the verb can be learnt. Then, he is asked to use this word in a sentence, to describe the action that has just been taught. At this point a grammar allowing all possible constructions (even those that are semantically wrong or that use unadapt prepositions) is used to recognize the description coming from the human. The exact sentence said will be stored and used later on for synthesis. For example, if I teach to the robot the action "Stéphane put the toy in the box", this very string will be stored. Then the arguments Stéphane, toy and box will be automatically extracted and stored as the "original" arguments, which will allow by simple string replacement to have the robot to describe "Peter put the tomato in the fridge". Indeed a "verb specific" grammar is somehow created online, without the need to define which type is the verb and how it should be used. This corresponds to the notion of grammatical construction as defined by Goldberg (1998). Moreover, since the sentence can be matched to an existing grammar, it is easy to assign to each argument a role in the sentence, which will allow generating other forms of spoken manipulation. The action put(Peter,tomato,fridge) can therefore be described, ordered or asked in any tense using the appropriate constructions ("Peter put the tomato in the fridge", "Does Peter put the tomato in the fridge?", "Was Peter put-ing the tomato in the fridge?").

It is also important to notice that the argument type is also important in the case of future recognition of the spoken description. Currently the system distinguishes clearly agents and objects, and they are stored in different vocabulary lists. At time of recognition this information will be used to generate a bit more accurate and restrictive grammar (which reduces drastically the risk of false recognition). In practice, the two actions initially described by "Stéphane put the toy in the box" and "Stéphane give the toy to Peter" will respectively produce the grammars "AGENT put the OBJECT in the OBJECT" and "AGENT give the OBJECT to AGENT". Although a neural or statistical learning system may link more robustly action verbs with an argument (Dominey and Boucher 2005) structured sentences, and the related link words, our representation can easily be integrated with any speech recognition software since for a specific action, a grammatically correct sentence can be

generated and that human users are most likely to produce those kinds of correct expressions.

Action Definition: Brain encoding and datastructure

Anatomical Networks

Grezes and Decety produced a synthetic review of imaging studies about action processing in human and monkey. They isolated the networks involved in different types of processing (motor execution, mental simulation, observation which embedded perception and recognition and silent verbalization) and described a cortical map of them which can be seen on (Grezes and Decety 2001). It is not surprising that many of the different processes use overlapping regions although the authors noted a relative independence of the silent verbalization network (which corresponds to our Descriptive Aspect of the definition).

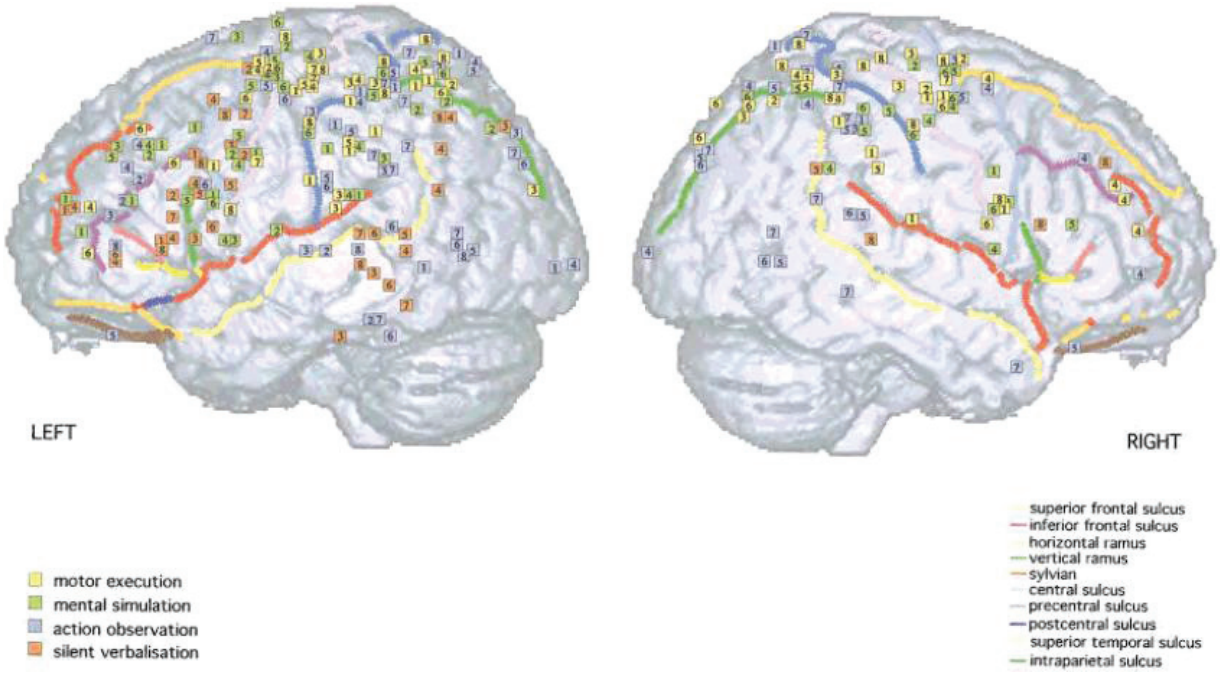


Figure 24: Extracted from (Grezes and Decety 2001). Show different cortical networks involved in the action definition and processing.

Naturally, real motor execution of an action and its mental simulation rely globally on the same pathway (which is mainly composed of motor and premotor cortices and of the DLPFC). Several authors ((Leonardo, Fieldman et al. 1995; Porro, Francescato et al. 1996; Roth, Decety et al. 1996; Lotze, Montoya et al. 1999)) reported a weaker activity in case of mental simulation, which may account for the hypothesis that a small activity of the action network can evok a mental definition, while a stronger one may lead to a physical execution.

Indeed the DLPFC seems to show a convergence of every process; it could be a good candidate for storing a relatively amodal symbolic representation of action. The fact the

Broca's area is included in this part of the cortex could explain our ability to manipulate those action concepts in grammar-like generative processes, and to translate these generated chains of actions to/from natural language.

In a paper about shared intentionality (I'll come back on this topic in the next chapter), Beccio and Bertone (Becchio and Bertone 2005) give an interesting summary of the action representation within its neural substrate. Of particular interest is their argument that the same representation is shared among the processes of production, recognition or imagination (and I would argue that it also encompasses description). They write: "At an intra-individual level, neural representations are shared in the sense that they are activated in different modalities of action. The same representation is activated when the subject executes the action, observes/hears another individual performing the same action, or simply imagines doing the action."

Within the context of modeling cooperation in robotics, Dominey and Warneken also studied the neural substrate of the action representation (Dominey and Warneken 2009), highlighting the role of BA46 in the ability to compose sequences of actions and claiming that the whole system can provide a "bird's eye view" (subject independent) representation of actions.

All of this anatomical evidence tends to show that it is possible to isolate an intricate set of cortical networks which deal with the whole concept of action and where different sub-networks are responsible for managing specific aspects of this concept. This is a key point that is also one of the most natural ways to implement all those processes on a machine. All of the possible action manipulations (observing, producing and verbalizing) are sharing a common structure while each of them is relying on a specific part of this data structure.

Implemented Datastructure

I gave in this chapter details about how the three aspects of an action have been implemented in our system. Although the definition is clear now and seems quite logical, it required a successive elaboration over duration of this PhD to deliver this stable, synthetic representation of the action concept in term of data structure. The Figure 25 shows a class diagram that has never been implemented in such a clean way; however it is the final one achieved after the process of ordering ideas through writing this manuscript. An action is

defined by three aspects: perceptual, motor and descriptive; however it is more than just linking perceptual events, motor commands and a sentence together: an action is a building block that can be chained with others in order to reach a goal. It can be seen as the basic element for teleological reasoning (cite our paper and one or more of the motivating papers on teleological reasoning), and therefore it should be characterized in term of conditions and consequences. We used to implement an action by specifying its "preconditions required, preconditions forbidden, post conditions added, post conditions removed". Those were basically the conditions and effects of an action upon the world state, they were part of the Primitive Action class which was a monolithic, stand-alone structure. However, I came to the conclusion that the three aspects mentioned above should be conceptually separated while all being linked within a higher level structure. This decision implies reconsidering where the "teleological material" (conditions & effects) are stored. What I will argue is that most of this is directly provided by the primitives (perceptual and motor) composing an action. Indeed, what one can do and the consequences it may have are immediately given by the physical world. My range of action is initially bounded to what my body can do given a specific physical state of the world; the consequences of my action will be detected as perceptual primitives characterizing the changes induced in this physical world by my motor primitive. However, all the teleological material that an action concept can embed is not restricted to this primary stage: higher level conditions and consequences can be specific to an action regardless of the primitives composing it. An example if the action "cover toy with box". The motor primitives composing it could be "*grasp(box), release (box,toy)*" therefore producing the perceptual primitive *(toy (isVisible 0)).*However, when I execute this action, despite the fact that the toy disappears from my sight, I know that the toy is still there, that it is "inside" the box. One could argue that this is a phenomenon described as object permanency in humans (Spelke 1990); nonetheless in our model it seems quite related to the action concept. Indeed it is quite logical to say that the action cover(toy,box) possesses the consequence *(toy (isIn box))*. This kind of properties belongs to the semantic level more than to the physical world state, however if we consider the world state in the broad sense it may include both levels. Moreover, having actions embedding this kind of semantic conditions or consequences allows to populate the world state with a semantic layer: when the robot recognizes an action (done by itself or another), then the semantic consequences are added to the world state, while executing an action can be constrained by certain semantic

conditions (e.g: I could grasp the box in any case, yet I could uncover the toy with the box only if the toy is inside the box).

If I had to identify the most important element of this thesis, I would say that it is this very concept of action and how it is modeled. Action is a building block for every robot behavior: a robot can execute an action, it can wait for another agent to perform it or it can speak and give/ask information about it. Every behavior can be seen as multiple action tracks going on in parallel, and we will see in the next chapter that cooperative abilities can be modeled using only a few more concepts in addition to that of action. This data structure should be seen as a tool, a tool is a useless artifact by itself, but it becomes very powerful if you are making the right use of it. In the last part of this chapter I will show a direct application of the action concept: modeling of the learning by imitation process.

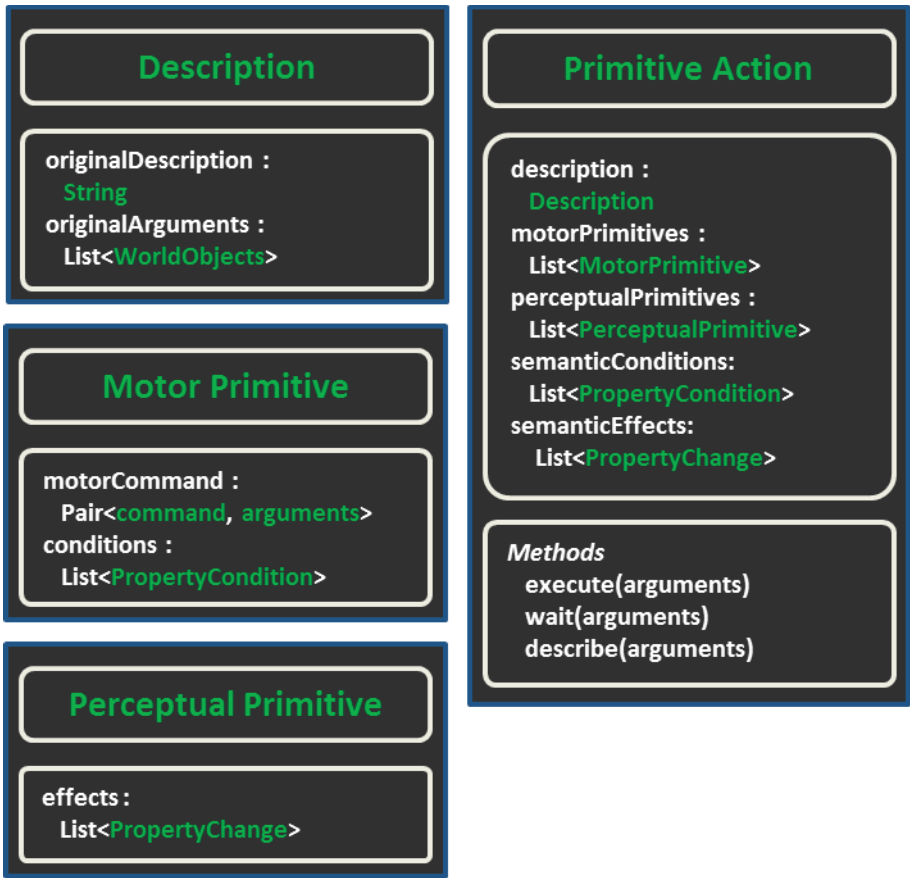


Figure 25 : Class diagram of the Action concept. An action can be executed, recognized or described.

Experimental Results: Application to Imitation

We already discussed the fact that children and adults have the ability to imitate actions of each other, and it is noteworthy that learning by imitation is a major area of research in robot cognition today (Alissandrakis, Nehaniv et al. 2002; Demiris and Johnson 2003; Dillmann 2004; Calinon, Guenter et al. 2005; Argall, Chernova et al. 2009). It requires being able to recognize an action produced by someone and reproduce it using our own motor representations. The action definition presented above allows naturally the robot to imitate his human counterpart: in ((Lallée, Lemaignan et al. 2011), Appendix 2) we described an experiment where the robot watches the user covering one object with another and then performs the same action.

Recognition Process Details

I have described in details the data structure used to store and manipulate action definitions within the architecture, I will now explain how the recognition algorithm performs on building and using this structure. After a learning phase, the robot is able to proceed to a previously demonstrated action as show on Figure 26.

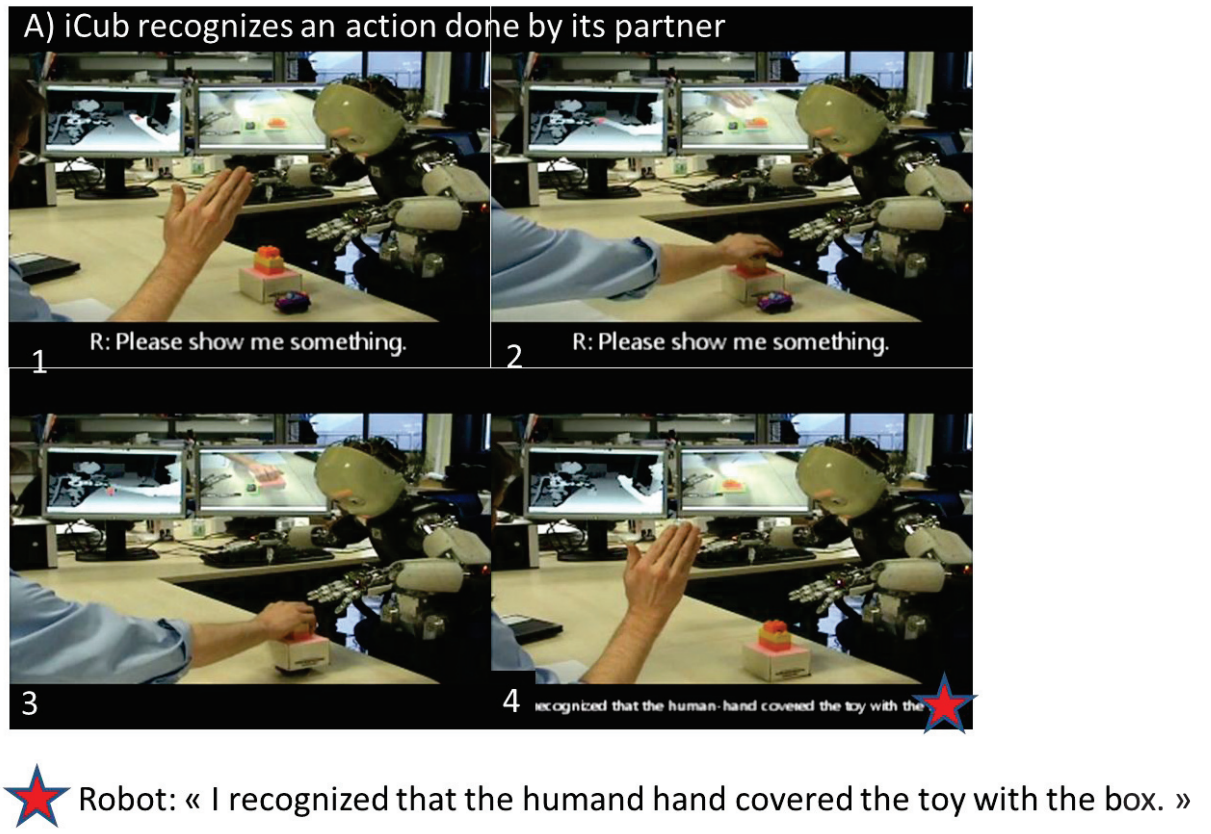


Figure 26: Imitation, observation phase.

Learning

In order to be able to recognize an action, first the system has to learn it. Although one could produce the appropriate definition by writing the representation apriori, the most natural way to train the system is to perform an action in front of the robot. First the robot is asked to pay attention to the environment by the spoken command "Watch", and then the user covers a small toy with a box. If we take a close up look to what happens in term of perceptual primitives we get the following sequence:

⇒ *Beginning of action*

⇒ Move(userHand, true)

⇒ Contact(userHand,box, true)

⇒ Move(box,true)

⇒ Contact(box, toy, true)

⇒ Visibility(toy, false)

⇒ Contact(userHand, box, false)

⇒ Move(userHand, false)

⇒ *End of action*

⇒ *2s without any perceptual change*

Those primitives are detected by the Primitive Recognizer module (see Figure 38 on page 120 for a diagram of the whole cognitive architecture), which is constantly monitoring the world state to detect changes. The Action Recognition module catches this stream of primitives and segments it in order to extract meaningful packets. When a primitive is caught, the segmentation process starts to record the stream until nothing comes in for a given time delay (2s in our experiments). This segment is used to build the early definition of an unknown action by calculating the state changed produced and setting appropriately the pre-conditions and post-conditions. I defined specific operators for primitives facilitate this

calculation: two primitives A and B can be summed and the result is a primitive C. The details for pre-conditions and post-conditions sum are given in equations (7)-(10) with:

- PrCRx being the Pre Conditions Required for x
- PrCFx being the Pre Conditions Forbidden for x
- PoCAx being the Post Conditions Added by x
- PoCRx being the Post Conditions Removed by x

$$PrCR_c = PrCR_a \cup PrCR_b \quad (7)$$

$$PrCF_c = PrCR_f \cup PrCR_f \quad (8)$$

$$PoCA_c = PoCA_a \cup PoCA_b - PoCR_a \cup PoCR_b \quad (9)$$

$$PoCR_c = PoCR_a \cup PoCR_b - PoCA_a \cup PoCA_b \quad (10)$$

The sum of all the primitives composing the segment is calculated, and the result is a new primitive which will serve as the basis for the new action. In our example the new primitive effects are:

Post Condition Added: contact(box,toy)

Post Condition Removed: visibility(toy)

This partial action definition is then processed by the recognition algorithm, and if no action is recognized the user is asked to provide details about what he did in the form of a spoken sentence (e.g. “Stéphane covered the toy with the box”). This sentence is parsed and synthesized in the form *(subject, verb, object1, object2).* Then the algorithm names this action, goes through all the pre-conditions and post-conditions, and it replaces the objects names by their role in the sentence (in our example the action adds *contact(object2,object1)* and removes *visibility(object1)*). At this point the action definition is ready; however the user is given the possibility to “edit” it by adding more pre or post conditions using the following grammars:

“If I want to *action* the *object* [with the *object*], then the *object* needs to be *relation* [with the *object*]” (pre-condition required)

“If I want to *action* the *object* [with the *object*], then the *object* should not be *relation* [with the *object*]” (pre-condition forbidden)

"If I *action* the *object* [with the *object*], then the *object* will become *relation* [with the *object*]" (post-condition added)

"If I *action* the *object* [with the *object*], then the *object* will no longer be *relation* [with the *object*]" (post-condition removed)

The action is then added to the recognizer database so that next time it is observed the system will recognize and describe it.

Recognition

After the segmentation has occurred, the recognizer tries to match the unknown action with existing action definitions in its database. The problem is to compare two action definitions based on their pre-conditions and post-conditions, in order to determine if they are equivalent or not. The recognition algorithm we used is quite simple, while not extremely robust. We define a similarity measurement between two actions based on the differences of their effects, and then when the recognition occurs we go through all actions of the database and calculate their similarities against the current action. All actions for which similarity is above a certain threshold are accepted as candidate templates for being recognized. The distance measurement is given in equation //ref.

$$S_{ab} = 100 - (\#(PoCA_a \,\Delta\, PoCA_b) + \#(PoCR_a \,\Delta\, PoCR_b))$$

A similarity of 100 means that the two actions have the same consequences on the world: all the relations they add or remove are the strictly the same. Each relation that is added or removed by one action and not the other will make the similarity decrease. All the actions of the database which are similar enough are stored, and for each of them the pre-conditions (required and forbidden) are searched for in the world state. If any required condition is missing or if any forbidden condition has been found, the action is discarded and cannot be recognized. In the end the most similar of the actions left is said to be the one which has been recognized, its arguments are set to match the ones from the perceived action and the action is described by the robot using spoken sentences of type:

"I recognized that the *subject action*-ed the *object* [with the *object*]"

Execution Process Details

After the recognition occurred, if the robot is in imitation mode, it will ask to the user whether it should execute the same action. When the user acknowledge, the robot will look into the action definition if the motor command to execute this action is known or not. Here again a learning process can occur, allowing the robot to learn what to do in order to execute the given action. After this learning phase, the robot is able to reproduce any action it has previously learnt to recognize (see Figure 27 as a sequel of Figure 26).

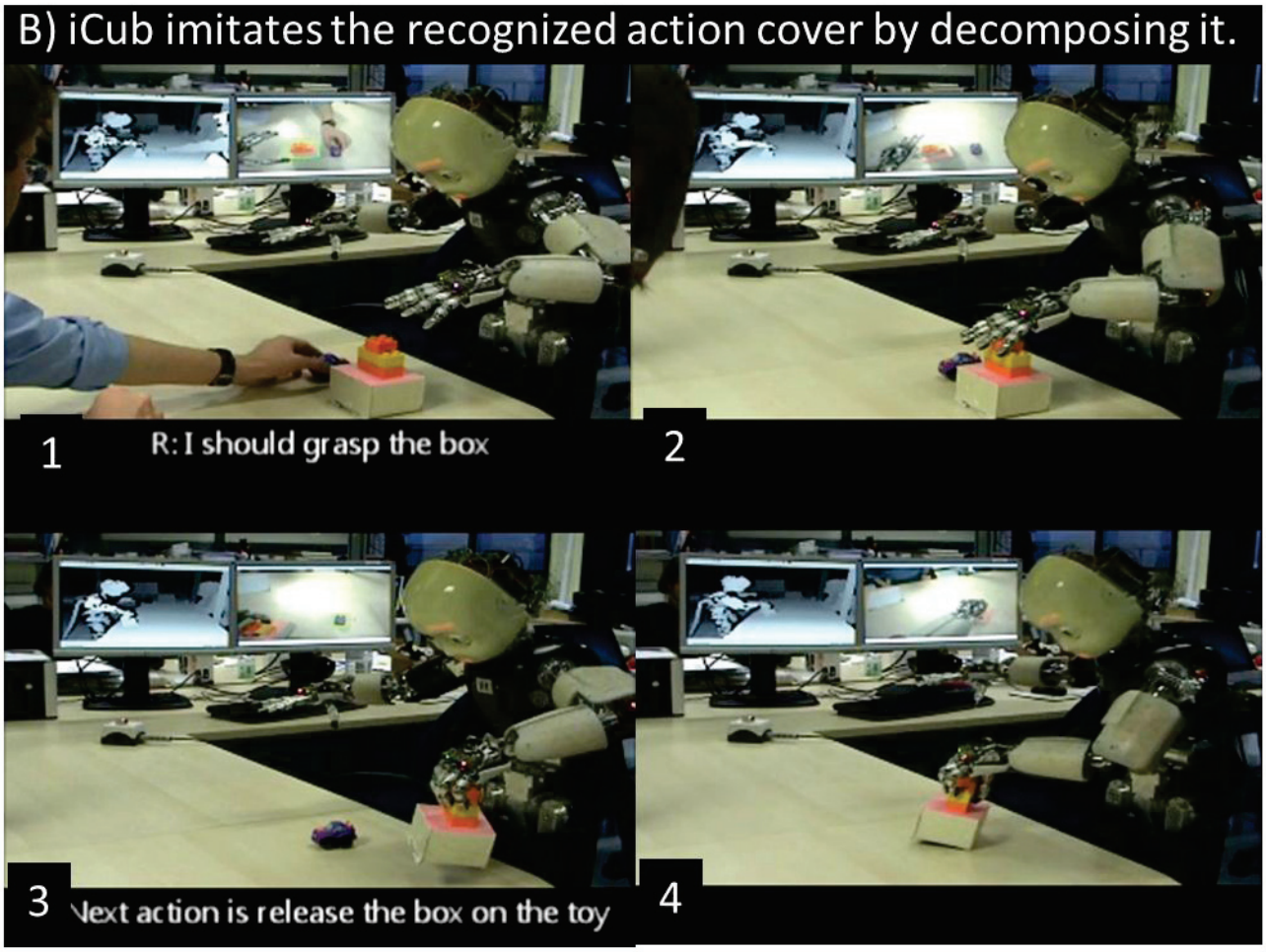


Figure 27: Imitation of a previously recognized action by executing each motor primitive which is composing it.

Learning

The motor sequence for an action is indeed a list of motor primitives as described in Action Definition: Motor Level. The robot will ask to the user what to do and expect primitive commands of type:

- *Grasp (object)*

- *Release (location)*
- *Touch (object)*
- *Look-At (object)*

When the user finished enumerating the successive primitives to perform, he can say "Finished" and the robot will update his action definition with the motor part. During the process a mapping of primitives arguments to their role in the sentence occurs, so that the *grasp(toy)* of the action *cover(object1,object2)* will become grasp*(object1).*

The learning process for motor commands may appear somewhat artificial in the last experiments; indeed it required to teach by language the sequence of primitives composing an action. Due to time constraints we implemented only direct spoken interaction learning. If we consider each motor primitive as a non-composite action that can be recognized, therefore it is possible to extract directly the list of primitive which should be achieved in order to execute a recognizable action. We presented in a previous experiment that such a learning can be achieved using observation (Lallée, Warneken et al. 2009), while this has been realized in the context of shared plans, the same principle applies to composite actions: their motor component can be learned by adding those primitives of which they are composed. Indeed we will see in

 that all the mechanisms are already in place to use this kind of learning.

Execution

When the motor part of the action is known, it is straightforward to execute it. The robot goes through the list of primitives and sends each of them sequentially to the motor command module. This module is robot specific and its only role is to implement the motor

execution of the primitive pool. When a primitive has been sent, the robot will await its end before proceeding to the next one. Note that during the execution, robot perception is likely to be corrupted by its motion (i.e. many object false recognition may occur, and false perceptual primitives will be triggered). This is problematic because we sometime want the robot to be able to experience the world by itself, for example by trying the execution of a random sequence of primitives and learning at the same time the perceptual outcome. To avoid this problem, when an action is executed the perception process is modified: the world state is recorded when the execution starts and again when it ends. By performing the subtraction of those two world states the robot gets the perceptual change (in terms of perceptual primitives) that the motor sequence produced.

Discussion

In this chapter I presented the data structure defining an action and how it is manipulated both in term of perception, execution and verbalization. Actions are characterized using primitives, this allows a symbolic representation which is easy to handle, modify and interpret. However this gain in clarity results in a loss in robustness. Many of the mirroring skills demonstrated in the literature(Johnson and Demiris 2005; Metta, Sandini et al. 2006) use the perceived motor state of the agent (i.e. its kinematic evolution over the action) to both recognize and execute actions. This has been combined with goal-based representations (Calinon, Guenter et al. 2005). Our system is based on the fact that each action can be recognized by its perceptual consequences in the world state (object states) and then performed by executing the associated motor commands. Those motor commands are not robot specific, but the primitives they call are, which implicitly solves the correspondence problem described in (Alissandrakis, Nehaniv et al. 2002; Nehaniv and Dautenhahn 2002). Although we cannot argue that our system can cope with the same range of actions as a "trajectory based" systems, it is complimentary with such systems, and can be used at a higher level, for actions involving multiple arguments and symbolic goal achievement more than precise motor imitation. Indeed, this approach also emphasis the equifinal means of an action since the user can demonstrate an action and then the robot will achieve the same result with completely different trajectories.

Aspects of this work can thus be considered in the context of learning by imitation or demonstration, which is a major area of research in robot cognition today (Alissandrakis, Nehaniv et al. 2002; Nehaniv and Dautenhahn 2002; Calinon, Guenter et al. 2005; Johnson and Demiris 2005; Metta, Sandini et al. 2006). Our novel contributions to this domain include (1) the encoding of action in terms of perceptual state changes and composed motor primitives that can achieve these state changes, in a manner that allows the robot to learn new actions as perception – execution pairs, and then use this knowledge to perceive and imitate. (2) These actions can take several arguments, e.g. *AGENT put the OBJECT on the RECIPIENT*, which allows for the generalization of learned actions to entirely new contexts, with new objects and agents. This yields the equifinal component of action where the same goal can be achieved by different means. (3) We use spoken language interaction and visual perception to provide learning input to the system. In our long term research program, this

provides the basis for learning to perform cooperative shared tasks purely through observation.

In our system actions are encoded using the effect they produce on the state of the world, the latter being abstracted in terms of unspecific quantities like relative position and orientation of objects and their visibility. The particular type of encoding we adopt for actions is therefore completely independent of the robot platforms, and can be transferred between robots with different embodiments or perceptual systems as we will see in Chapter V.

Chapter III

Cooperation, using Actions to compose Shared Plans

Introduction

Collaboration is one of the hall-marks of human social life. By pooling their efforts in joint cooperative activities, people can produce outcomes that lie beyond the means of individuals, reaching from simple acts of lifting heavy objects together, over hunting in a group to the building of towers. One important aspect of collaborative activities is division of labor and the assignment of who performs which role in the joint activity. Indeed, it is likely that one of the other hall-marks (human language) evolved in part in order to support the human ability and need to organize cooperative behavior (Tomasello 2009). Here we consider collaboration in terms of two individuals who have a *shared plan* that involves actions performed by both, in a structured temporal sequence, in order to achieve a *shared goal* which is the desired outcome of their *shared intention*. The previous sentence purposely used an amount of undefined vocabulary: intention, goal, plan - all of these can be qualified as being "shared" and those notions have been defined previously in the literature. I would like to first introduce those terms that I will use intensively in this chapter; however, looking at the literature it appears that just giving such a definition is already the matter of a complete thesis. The definition of shared intentionality has been debated extensively (Cohen and Levesque 1990; Bratman 1992; Bratman 1993; Velleman 1997) and some philosophers are still arguing whether it is possible to share an intention. I base my work in part on an existing framework about intention, goal and planning in the context of cooperation as defined by Tomasello and colleagues (Tomasello, Carpenter et al. 2005). In this context, as developed in the previous chapter, the core advancement achieved by this thesis was to give a synthetic definition of action that is effective in the domain of human-robot cooperation. Acting can be achieved within the context of a plan, a shared plan or a single action, but in all of those cases the main question stay "which action should I choose given the current state of the world and the state I would like to get closer to?" (With the "state of the world" ranging from the internal mental state of the subject to the mental state of others passing by the physical state of the environment). Therefore, it is the very concept of action which will drive my dissertation on cooperation and shared plans and I'll try to show how such concepts can naturally emerge from a single action. The notions of goal and intention are indeed quite intuitive, and this may be the reason that makes their formalization so difficult. For the following chapter I will consider a goal as being the state of the world that an agent wants to reach, and an intention the actual will to follow a plan

toward this goal. One of the main questions animating the debate going on between philosophers is if an intention, a goal and a plan can be shared amongst multiple beings. Indeed it has been argued by Tomasello et al. (Tomasello, Carpenter et al. 2005) that human is the only animal species which demonstrates the ability to cooperate while sharing an intention and a goal. Moreover, children appear to show an intrinsic desire to share mental states and intentions of others. Despite the fact that my mentors probably biased me regarding the existence of this shared intentionality, I will try to study and model cooperative activities in a neutral way along this chapter. However, the last part of this chapter will show through a human robot interaction experiment that shared intentionality is requirement for successful cooperation.

Let's call the roadmap of multiple beings acting together toward the same goal a shared plan. Shared Plans can be observed in most of the activities which involve more than one human being. They range from low level sequences of motion, performed by two partners in an organized and relatively simple rule based reactive system (e.g. dancing), to high level strategies performed by thousands of people working toward the same goal by accomplishing a hierarchy of tasks (e.g. military strategy, economical strategy etc.). While it is common to classify actions according to the fact that they are goal directed or not, it appears that Shared Plans are always considered as being a way to achieve a shared goal, with a goal being a particular state of the world, or a modification of this state. I do agree with this view, although in the case of a Shared Plan like dance, the goal is something more abstract than the physical world state. The notion of Shared Plans has been initially formalized by Grosz and Sidner (Grosz 1988). More than a succession of actions attributed to agents, it models how intentions, beliefs and sub-goals evolve during collaboration. In their paper from 1988 the formalism was intended to model Plans shared by two individuals, however Plans can be shared within larger groups which led them to extend their model (Grosz and Kraus 1993). In the computational world, shared plans theory of Grosz has been used as a basis for implementation of artificial collaborative systems. Collagen (Collaborative Agent) is one of those systems, which controls the behavior of a virtual agent interacting with a computer user in order to help him using a program (which in some case simulates a real situation). The gap between such a system and human robot interaction is indeed very small: the virtual agent of their system and the user are interacting through speech,

manipulation of the interface and pointing. If we replace the graphical interface by a physical setup, then all the theory applies directly to human-robot interaction. A schematic view of the interactions between the agent and the user is represented in Figure 28. The core capabilities required for collaboration appear clearly: each agent must be able to perform actions, to recognize actions and to communicate with other agents. Communication is not considered as an action because it is mainly used to supervise the shared plan execution (to gather/provide information). Since all those requirements are embedded in our action definition (Chapter II), it can be directly applied to Shared Plan management.

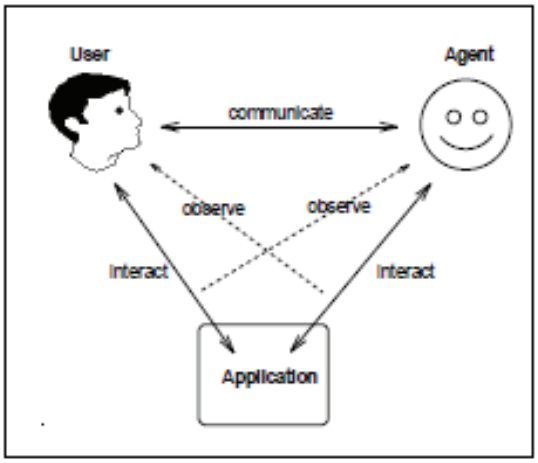


Figure 28: Collaborative interactions overview. Extracted from (Rich, Sidner et al. 2001)

Shared Plans: Neurophysiology

Studies about shared plans and intentionality within the neural substrate are tightly coupled with studying the mirror system. As mentioned above, plans (shared or not) are sequences of actions that are surrounded with notions of intention and goal, which means that the neural network encoding the notion of Action is likely to be involved in the one dealing with shared plans. Indeed Becchio and Bertone raise an interesting point in (Becchio and Bertone 2005) by observing that since the brain is using the same cortical network for both production and recognition of action, then the main problem is not how to have a shared representation of an action with someone else, but to differentiate between our own actions and those of others. Sharing a plan with someone else raises a difficult problem: how to distinguish between self and other? Apart from metaphysical considerations, distinguishing between myself, other A and other B, consists mainly in being able to assign the right agent to each action observed or imagined (note that a self-produced action is also observed). Indeed, brain imagery provides important insight about how it is possible to make this distinction in human beings: several PET studies (Ruby and Decety 2001; Farrer, Franck et al. 2003) demonstrated that neural networks involved in mental simulation of actions done by self and others are overlapping : the superior temporal sulcus, the medial prefrontal cortex and the inferior parietal lobule are activated for both observed and initiated actions. However, the inferior parietal lobule has a lateralized activation: the left part seems to be assigned to actions done by the self, while the right one would code for actions done by others (Chaminade and Decety 2002; Decety, Chaminade et al. 2002; Becchio and Bertone 2005). It is important to note that those studies are done on the case of imitation in order to keep the same baseline for the action and to distinguish between self and other. However what is particularly interesting in those studies is that if self and other have two distinct statuses, it means that only the self is not encoded like all other possible agents. When I observe A or B producing an action, then the “others” area will be activated, producing the important information that this action is not mine and the treatment of “who is this other” will be done but of less importance for the action. When using an action concept (whether to observe, imagine or describe) there must be an initial test: is this action related to myself? The processing of the action symbol will be completely different if the agent is me, while it

will be similar among other agents. It is something that is quite straightforward and that we will retrieve also on the implementation side.

Another characteristic of shared plans is that they intrinsically require the "we-mode" defined by Tuomela (Tuomela 2001; Tuomela 2005). Searle (Searle 1990) noted that a concurrent activity of people which may look like a shared plan may not really be defined as a cooperative activity. Indeed he give the example of people sitting in a park, then the weather starts to rain and everybody stand up and looks for a cover. However it is not possible in any case that those people have a kind of cooperative behavior, they are just acting on their own, and the global picture of their concurrent actions may look like cooperation. The "we mode" involves that the agents share the desire to accomplish an action or a plan together more than in an individual way. The neural basis of this phenomenon have been investigated in a review by Becchio and Bertone (Becchio and Bertone 2004) which led them to conclude that "*in different areas of the brain neural representations underling the self and the other's behaviour share a common, we-centric code*." Indeed, the term "we centric" code has been defined in one of their previous papers (for Italian readers : (Becchio and Bertone 2002)) and used as a basic principle in (Gallese 2003), it sums up the idea that the cortical representations of actions are mostly independent from their subject. Indeed this "we-centric" term highlights the fact that the agent is only another parameter of an action, the brain structure coding an action stay the same while all the possible parameters may be stored in another area and just "linked". In Figure 29 I give a schematic view of my understanding of this phenomenon, it is applicable both for the cortical organization and for my implementation within the robotic system. The hierarchical structures Primitives, Actions and Plans are completely abstract from Agents or Objects which are just linked as arguments (parameters), they are therefore "we centric" in the sense that the same structure, cortical code, can be used both for an action done by myself and by others. Given that a plan is a parameterized sequence of actions, the "we mode" can be seen as a composition involving some actions done by the self and some done by others.

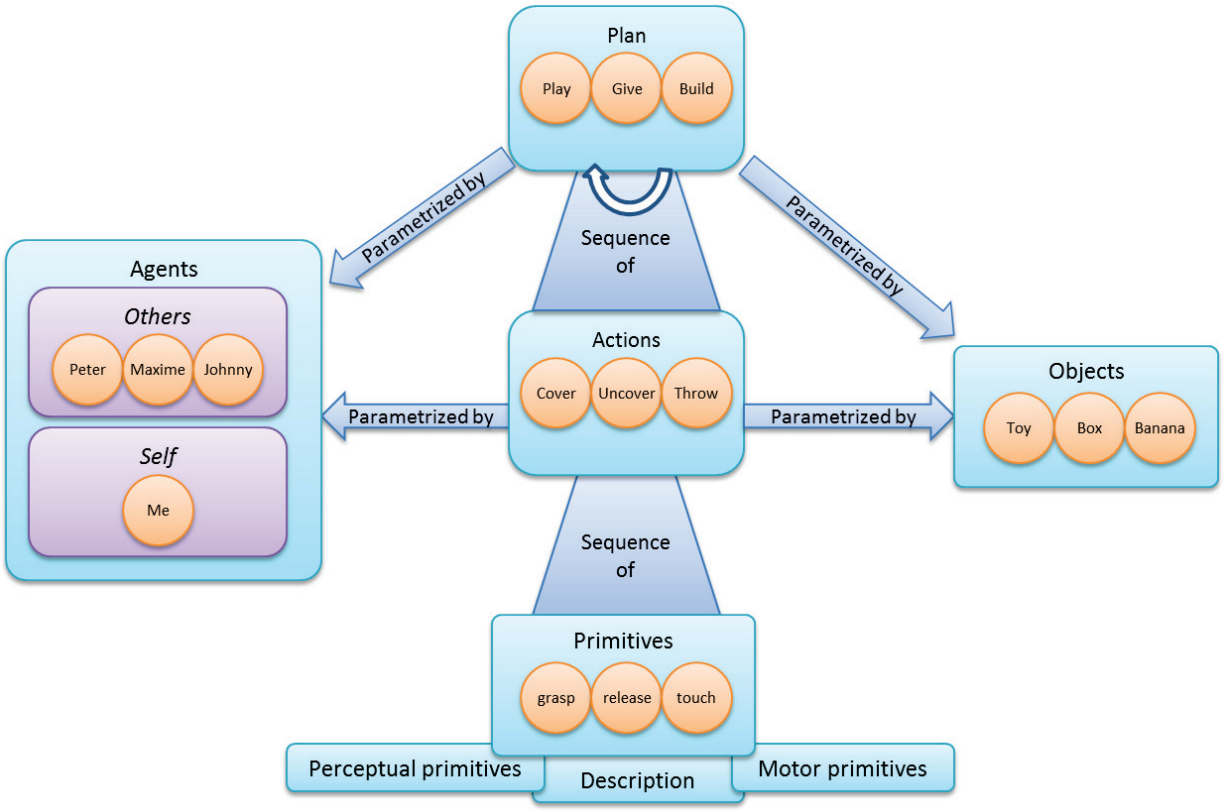


Figure 29: Cortico-inspired model for shared plans

This self/other consideration, and more generally speaking the attribution of agency to an action; relies more on the action concept than on the plans definition. As I said a Plan is a goal directed sequence of actions, those actions may depend on some specific conditions, on the people presents, on the objects available, etc. Indeed again we can identify the parallel between planning and language: action is to sentence as plan is to paragraph. Plans are hierarchical organizations of actions in a grammar like format; therefore my expectation would be that Broca's area is involved in the manipulation of plans. A very good overview of the role of Broca's area is given in (Hagoort 2005) from which I took Figure 30 ; the paper describes a framework to model hierarchical speech processing and production, most of their ideas directly apply to the case of shared plan comprehension and execution. The key point is the process of "Unification", how to combine multiple elements into one bigger concept. We already seen this notion when talking about actions composed of multiple primitives; the same mechanism is present in building plans, as described in Figure 29 a plan is a composition of multiple actions and/or plans (it is a recursive structure). When dealing with action perception we already noted that the left inferior frontal girus (LIFG) was

involved, it is also the case in speech unification (Hagoort 2005). Later work from the same authors (Hagoort and Van Berkum 2007; Willems, Özyürek et al. 2007; Hagoort, Baggio et al. 2009) show that this network is also involved in the binding between language and action, and that it more generally deals with unification processes in a generic way. For Hagoort, unification is one element of a three parts model called MUC (Memory, Unification, Control).

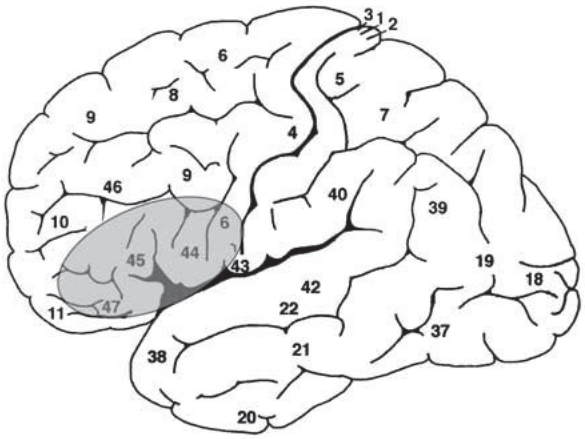


Figure 30 From (Hagoort 2005) Lateral view of the left hemisphere. Lateral view of the left hemisphere. Brodmann's areas (BA) are marked by number. Classically, Broca's area comprises BA 44 and BA 45. Adjacent language relevant cortex also includes BA 47 and ventral BA 6 (grey oval).(Decety and Sommerville 2003; Becchio and Bertone 2005; Knoblauch, Markert et al. 2005)

Shared Plans: Child Development

Learning a shared plan by observation

"Human children at 14-24 months display a remarkable ability to observe adults perform a cooperative task (with only 1 or two demonstrations) and then to engage themselves in that task, taking the role of either of the demonstrating adults (Warneken, Chen et al. 2006; Warneken and Tomasello 2007). Tasks typically involve retrieval of a toy from a physical device which requires both agents to manipulate it in a temporally organized and synchronized manner. By definition, the goal directed tasks require two agents to collaborate – as the physical constraints of the task are such that an individual agent cannot achieve the goal. The behavioral data indicate that the children have understood the task in terms of a coordinated succession of actions, rather than a set of specific motor trajectories. This research has identified three principal characteristics for collaboration (1) agents are mutually responsive and coordinated in their actions; (2) they have a common shared action plan for the joint enterprise. (These provide a "birds eye view" of the collaboration and can be demonstrated by the agents' ability to reverse roles.), and (3) a mutual commitment to subsume their individual actions to the joint goal (Tomasello 1999)." (from (Lallée, Warneken et al. 2009))

Cooperation in its most simple form occurs early in child development. From soon after birth toddlers are already able to coordinate their interaction with another person in a turn-taking way (Trevarthen 1979). This early turn-taking ability may be a basis for later coordinated execution of shared plans. From 6 months those coordinated interactions can become triadic by involving manipulation of an object (Tomasello 1995), however the cooperative games at this age seems quite "frozen" and the child cannot generalize the game principle over different objects/agents until he his 18 months of age (Hay 1979; Ross 1982). Starting from 20-24 month, children are able to extract individual's actions in terms of their object manipulation goals and attribute these to the appropriate agent, forming a "bird's eye view" of the collaborative action. Warneken, Chen and Tomasello (Warneken, Chen et al. 2006) have studied several cooperation situations where two individuals have to reach a common shared goal which is impossible to achieve alone. In one of the situations, two children are confronted to an unknown "transparent chest" device which is locking a toy. Their goal is to retrieve the toy lying inside, however to do this one of them has to activate a mechanism

which will make the toy accessible to the second one. The experiment occurs in two phases: first two adults demonstrate a successful shared plan to realize this joint task, and then the children have to act on their own. As we said, children of 24 months are able to build a Shared Plan representation based on the adults' demonstration: after one or two observations of the plan, they are able to engage it with a partner in any of the roles, thus executing the same sequence of actions they observed with the role of agents being a parameter they can change according to the situation. This generalization capability appears at 20-24 months, when children becomes generally capable of generating coordinated acts in non-ritualized contexts (Eckerman 1993). Being able to learn a plan by observation of two agents cooperating implies that the child understand several notions:

- He understands who is the agent performing the action, and that this agent is the same over multiple actions (agency attribution)
- He understands that the two agents performing the plan have a special relationship and are not acting on their own but in a collaborative way (shared intentionality detection)
- He understands that the first actions need to be achieved in order to allow the following parts of the plan to occur (causal relations along the sequence of action implementing the shared intentionality)

Execution of a shared plan

A shared plan is the commitment of multiple agents to achieve a predefined sequence of actions which is aimed at moving the world state closer to their goal. The plan can be generated by learning or by social elaboration through dialog (spoken or not). Both ways will end up in the creation of a representation of this plan which will be shared among the agents who are committed to it. Before this exact roadmap is generated, agents share the intention to reach their shared goal, while not having yet a plan to achieve it, this is call Goal Intentionality. Whenever the plan is produced and the participants agreed on achieving it when the initial conditions are met, this Goal Intention turns out into an Implementation Intention.

Goal Intentions and Implementation Intentions are concepts that has been defined by multiple researchers over the last century (see (Gollwitzer 1993) for a review ; see (Ach 1935; Lewin 1951) for historical basis). Tomasello's group have led several experiments on children using such shared plans by cooperating in order to play a game or reach a reward (Warneken, Chen et al. 2006; Hamann, Warneken et al. 2011). Apart from the initial implanted intention, participants are always monitoring the status of others in order to adapt the plan in case of difficulty or to share information about the progression status through gaze, body and spoken language. When executing a shared plan, one should not just execute in a "brute" way all the actions he has attributed, instead he should be synchronized and coordinated with his partner, by recognizing their actions, sharing their attention and informing them about his own state. As we will see in a further described experiment on naïve subjects interacting with a robot, those behavioral components are somehow required in order to involve the partners into the shared plan execution.

Shared Plans: Implementation

A significant open challenge in human-robot interaction is how to transfer task knowledge from the humans to the robot. This is particularly challenging in the domain of collaborative interaction in which the robot and human should take turns in a structured shared plan as seen in previous work (Dominey, Mallet et al. 2007; Dominey, Mallet et al. 2007). Interestingly, human infants display a remarkable capacity to learn collaborative behavior from a single demonstration, and to use this knowledge to take either agent's role in the collaborative behavior; implementation of this behavior provides the robot with a powerful way to learn from observation.

Learning a shared plan by observation

Our desire was to mimic the ability of children to learn a shared plan by a single observation of other agents' performance. The task was to give this ability to the robot as a new tool for learning by demonstration. Given the fact that the system possesses a capability to recognize an action, including arguments and agents, learning a plan by observation is quite straightforward. The setup experiment that we presented in (Lallée, Warneken et al. 2009) was composed of a two handed box (which cannot be lifted with only one hand) and plastic toy. First the robot had to observe two humans (Larry & Robert) performing the task: one of them lifted the box, then the second one was able to grasp the toy and finally the first one released the box on the table. After that the robot was placed on one side of the table, in front of a human with the box covering the toy on the table and asked to execute the plan demonstrated.

Three software modules are involved while learning from observation:

- Action Recognizer is used in recognition mode to detect which action is done and who did it
- Spoken Interaction is used to confirm the action recognition results and to filter the communication between Action Recognizer and Shared Plan Manager
- Shared Plan Manager's role is to manage Plans, which includes atomic actions recognized by Action Recognizer and more complex hierarchical arrangements of those actions. It is responsible for creating new plans and using information coming from the Action Recognizer.

First the Shared Plan Manager is instructed that a new plan will be learned, this plan is given a name and the modules starts to listen for action definitions as input. During the demonstration, the Action Recognizer detects the three atomic actions:

- Lift(Larry, box)
- Grasp(Robert, toy)
- Release(Larry, box)

Each action definition is broadcasted and the Spoken Interaction module catches and expresses them so that one user can confirm the perception (e.g "I saw that *Larry lift*-ed the *box*. Is it right?"). While the scenario could be more natural and fluent without this confirmation phase, the recognition system is too sensible to perceptual noise to operate without user interaction. The validated actions are sent one by one to the Shared Plan Manager, which is instructed to append them to the new Shared Plan. Each step in a plan is called a sub-plan and those sub-plans are recursively stored in list of plans, thus allowing hierarchical definitions. A requirement for this system to work is that each recognizable action (i.e actions that are known by Action Recognizer) has a corresponding atomic plan in Shared Plan Manager; indeed any plan can be part of the sub-plans list, however the hierarchical leaves have to be plans corresponding to atomic actions.

Execution of a shared plan

The first step in a shared plan execution is to actually share the plan and agree on the respective roles of each agent. In our case there was not any question regarding which plan was about to be executed

Generation of a plan: teleological reasoning

Our action definition can be understood as a function which adds or removes relations between objects in the world state. Given a sequence of actions (a plan) *P* which will change the world from a state *S1* to a state *S2*, we name this modification *dS*. Although our actual definition of *dS* could be understood as purely perceptual it can also be seen as a goal. If we assume that *dS* are the effects of a goal-directed plan, considering that it is also a path from *S1* to *S2*, then we can establish that:

1) Reaching S2 is the *Goal Intention* associated with P. P being one of the multiple equifinal plans which can lead to S2.
2) The planned execution of those actions when their pre-conditions will be verified is an *Implementation Intention*

This framework is consistent with our action definition and thus can help us to formalize the robot wills and acts. What we called a Plan corresponds indeed to an Implementation Intention, however any plan possess an underlying Goal Intention. When observing the effects of a plan it is possible to extract the goal intention which motivated the agent to execute it. It is also possible to record that this specific Implementation Intention was one of the possible ways to reach this goal state assuming a specific start state given by the current context.

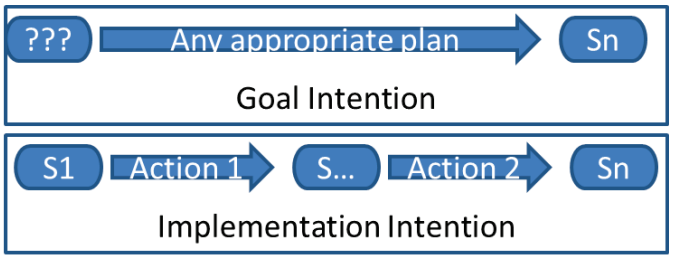


Figure 31: Different intention types. Goal Intention represents the commitment to reach end state Sn without a specific plan or starting condition. An Implemented Intention represent the commitment to reach Sn using a specific plan whenever the state S1 is observed.

Within our implementation, we can get the system to express the goal associated with a specific plan or action by verbalization of the post conditions. Let say that a plan P is composed of the actions [*uncover(robot, toy, box) ; give(robot, toy, human)*].

Uncover(toy, box) will produce the following changes:

- Add visibility(toy)
- Remove is-covered(toy, box)

Give(toy, human) will furthermore change the world with:

- Remove visibility(toy)
- Add possession(human, toy)

If we sum all these changes, then what is remaining is that P breaks the covering relation from the box regarding the toy, and that it creates a possession relation from human toward the toy. Therefore the goals of this plan can be verbalized as:

“The robot had the goal intention to make the human to possess the toy” and

“The robot had the goal intention to have the box to not cover the toy”.

Of course the goal of the plan that we would naturally extract is only the first one (human possess the toy). Indeed the extraction process can be refined if we take into account the pre-conditions of each action composing the plan: assuming that give(robot,toy,human) will first implement a grasp(robot, toy), then its forbidding pre-conditions will contain is-covered(toy,any). If we remove from the exhaustive list the sub-goals (i.e. goals that were requirements for later actions), it is possible to reduce the goal intention of a plan to its simplest definition (in this case make the human to possess the toy). Being able to extract the goal intention from an observed plan is important to attribute concepts like intentions, desires and beliefs to the human agents. Although I didn’t have time to go that far and to push the experiments in this direction, goal attribution through action observation is a direct way to understand other’s mind and willing.

We’ve spoken earlier about certain experiments (Csibra, Gergely et al. 1999; Király, Jovanovic et al. 2003; Csibra 2008) that aimed to test the requirement for an action to be tagged as “goal directed” in infants. This implies the idea that not all actions are goal directed, however some sort of goal can be extracted from any action represented in my definition. Indeed being able to measure the “goal directedness” potential of an action could be a way to record and store only useful actions. Although the idea is interesting, it is difficult to implement “equifinality perception” in our current definition since the representation used is mainly symbolic while this core capability in infants is mainly based on meaningful trajectories of the agent. One way could be to say that an action definition of the database is tagged as a goal oriented action if and only if there is another action in the database which possesses the same *Goal Intention* (resulting state) but using a different *Implementation Intention*.

Being able to retrieve the goal of an observed plan is a useful feature, however a major question remains: how to produce an appropriate plan given a specific goal? This question has been central within the field of classical AI and planning is now considered as a standalone topic. The focus of this thesis is not task planning, however within the project architecture (see Chapter IV) we created an interface between this action and plan definition with a partner's planner, HATP (Human Aware Task Planner). Although the Action Definition format was not conceived with the goal of interfacing with an external planner, it was quite a bet to try to convert my plan definition to a HATP formatting. Fortunately, they were both grounded in the Shared Plans Theory framework (Grosz 1988; Grosz and Kraus 1993) thus the mapping between these two systems is quite direct. While planning is not the main focus of this thesis, we implemented all the bases required in order to produce teleological reasoning and therefore to extract, given an action or a plan, its consequences or its requirements. A snapshot of the robot cognitive system is given in Figure 32, it includes the representation of the action "cover" with its requirements and effects.

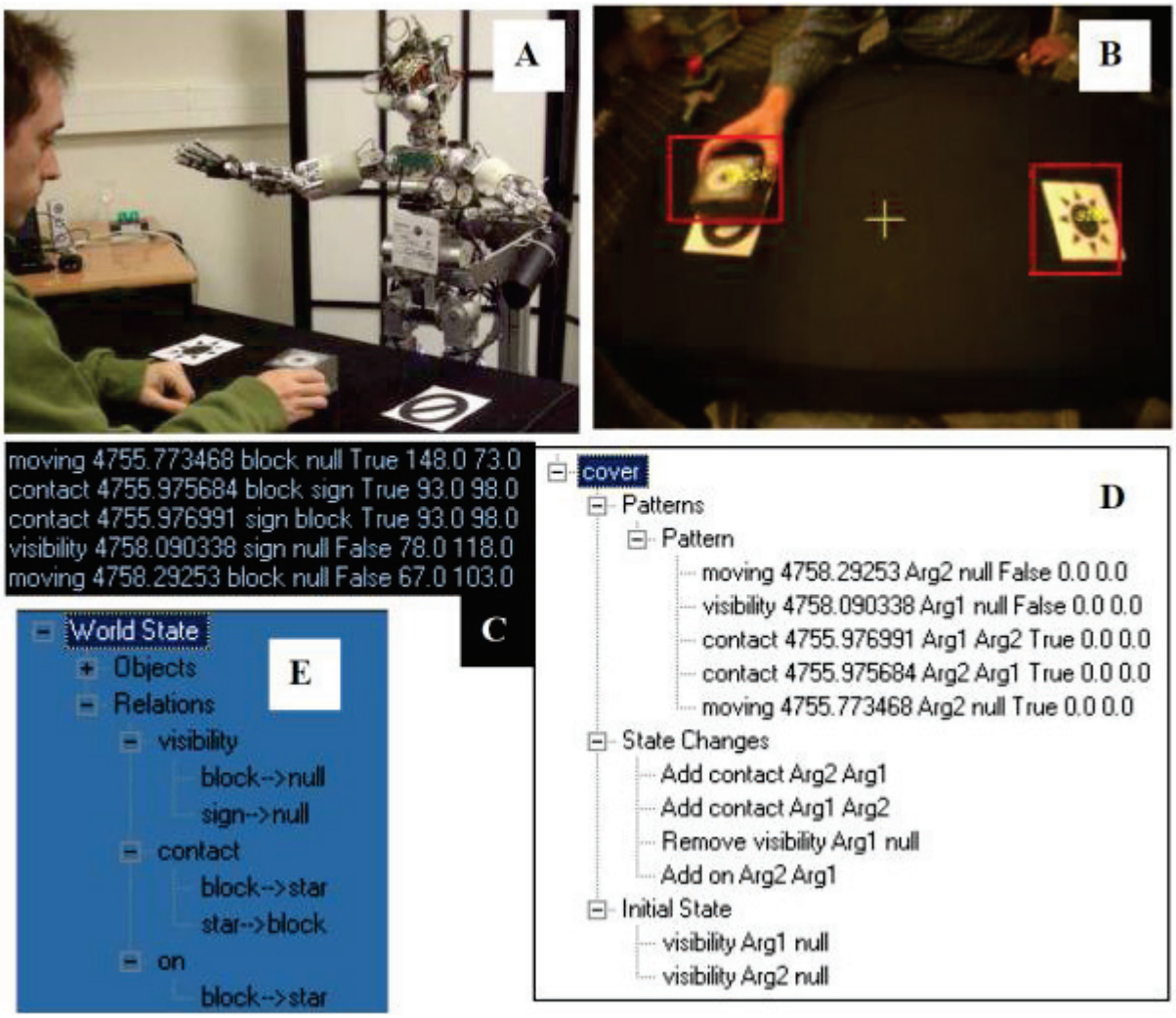


Figure 32 Learning and generalizing "Cover Arg1 with Arg2". A. Robot setup and visual scene before the action. B. Vision: Robot's view of scene after the block is put on the sign. C. Temporal Segmentation: Time ordered sequence of perceptual events observed during the action. D. Knowledge Base: Abstract pattern template for cover, along with the resulting state changes, and required initial state. E. Relations encoded in the World State after the "cover" is recognized in a new action "covers the star with the block".

Such a representation has been used to extract meaning from those conditions, allowing the robot to verbalize them in order to answer to some specific questions. Although at the time of writing those capabilities range are in single action, the extension to a plan is just a matter of backward/forward chaining. An example of the produced rules is given in Figure 33.

Robot : Recognized cover star with block

User : What are the conditions to cover the sign with star?

Robot : If you want to cover the sign with the star then

Robot : sign needs to be visible

Robot : star needs to be visible

The sign is visible, but the star is covered with the block, so the user asks the robot how to make the sign visible:

User : How can I see the star?

Robot : You can uncover the star

The user can now uncover the star by removing the block, and then achieve the final goal which is to cover the sign with the star

Figure 33 Interaction fragment where the robot performs basic goal-based reasoning about how to solve the task of getting the star when it is covered by the block, and then uses it to cover the sign.

Experiments

Teaching a shared plan to the robot and using to test naïve subjects

The current experiment examines how naïve subjects are able to cooperate with the robot in different experimental conditions that manipulate the shared plan. What does it means to share a plan with someone? Although the answer has been given in psychological terms, shared intentionality and commitment to pursue a goal are not notions that can clearly be implemented on a robot. It is something to notice how two people work together toward achieving something, about them transferring their own goal to the other using natural interaction and language, about adapting to a change in the environmental conditions, etc. However such behaviors are so natural to us that it is hard to not lose of sight most of its complexity. When individual A has to interact with B then it is required that both A and B express many cues to show their awareness about what is going on and about their partner state. I would like to come back to Figure 34, which has been taken from the "core paper" about shared intentionality (Tomasello, Carpenter et al. 2005), in order to come back to a so obvious fact that it is easy to forget: when X people shared a goal or an intention, it means that X different mental constructs for goal and intention are built. Every single agent possesses its own goal and intention, which are supposed to be the same or at least similar among all the population in the case of cooperation. A key factor in order to have the agent A to successfully go through the shared plan with B is that A believes that they are both sharing goal and intention, therefore that:

1. B has the ability to hold a goal and has the intention to reach it
2. B's goal and intentions are the same as A's

Subsequently B needs to somehow express his intentions and mental state so that A can trust they are both sharing a goal. From the robot side, it is hard to say that the robot is holding a goal or an intention: when it executes a plan systematically proceeded through a sequence of commands while monitoring its partner's activity in order to know when it should act, and when the shared plan is complete. In the current experiment, we manipulate the shared plan, such that in some conditions the robot attributes action to itself and the human, and in others, there is no shared plan, just a sequence of actions, with no specification of who should do what. As I said in the preface, the main question in human

robot interaction at this moment is how a naïve human will respond to the robot in these different conditions.

An agent has basically three ways to express its intentions and mental states: it can speak, gaze and physically interact with the world (including facial and body expression). If we want a robot to be attributed intentionality and goals then it needs to express them using those behavioral cues.

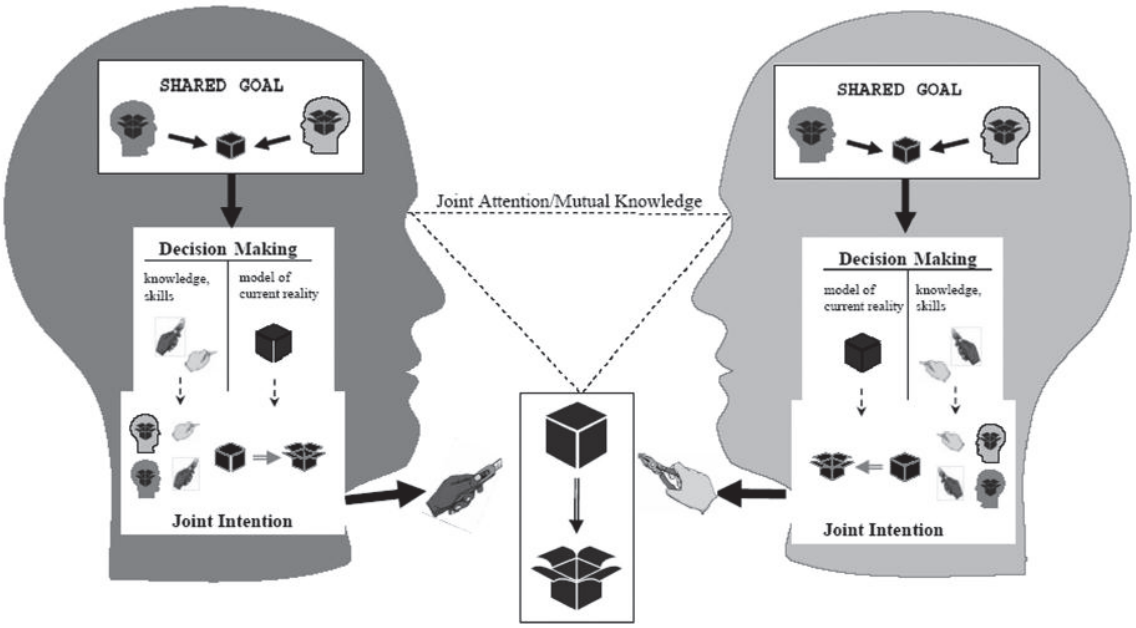


Figure 34 Taken from (Tomasello, Carpenter et al. 2005) Shared intentionality relies on sharing attention. Attention is the very first way to express our intentions.

The very last experiment conducted in this thesis was to have naïve subjects to interact with the iCub during simple cooperative games. Games were indeed shared plans similar to children experiments conducted by Katharina Hamann (Hamann, Warneken et al. 2011): the goal is to retrieve a toy hidden under a box through cooperation with a partner. In every trial agent A has to move the box, agent B take the toy and finally agent A replace the box where it was. As in every shared plan the role reversal is possible, so agent A can be either the robot or the human.

Our main focus in this experiment is to test how the means of expressing intentionality and mental state impact on the execution of the plan and on the interpretation of the human. Therefore we designed several conditions regarding this: intentions of the robot can be

expressed by speech, gaze or both. Concretely, at each step of the shared plan the robot is faced to an action which has to be executed either by him or by his partner. In every case the robot describes the action, for example if the current action is (uncover, human, toy, box) then the robot will say : “Now you uncover the toy with the box” while looking sequentially at the human, the toy, the box and back to the human. This will happen in the “full cues” condition, the spoken part and the gaze part can be separately disabled.

On every subject, several trials occurred in various conditions. At the time of writing the data is still being studied, however we were already able to extract from videos a major effect of gaze in sharing intentionality. Even without speech, the subject is able to understand the expectations of the robot. For example in Figure 35 he has no way to know in advance who’s the one going first, if he waits for a behavior of the robot he will be instructed either by the robot making the action or by the a succession of “gaze(box) gaze(subject)” (like a dog which is asking you to open the food can). The subject reacts really fast to produce the action.

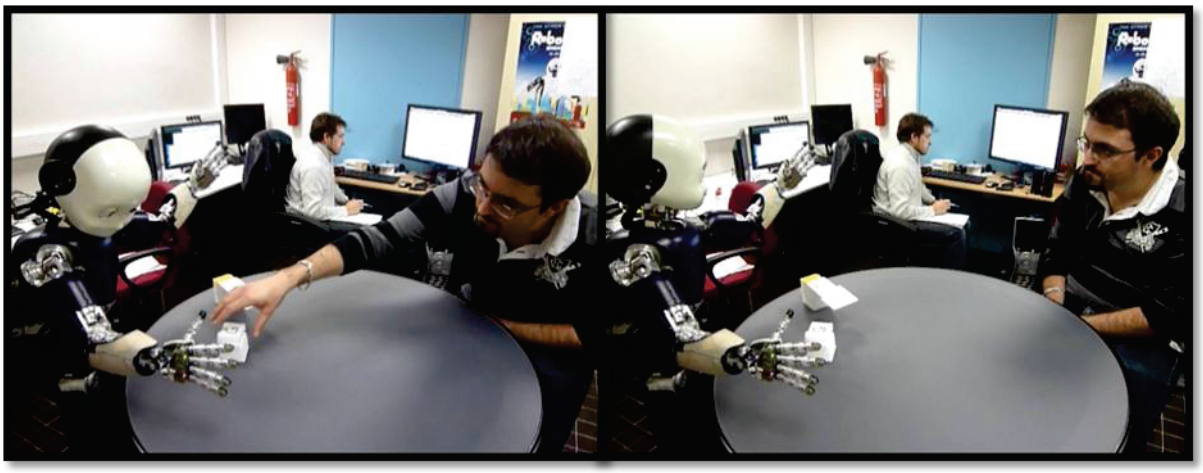


Figure 35 Left: In the shared plan, with gaze, no language condition, after uncovering the box, the robot indicates to the user “it’s your turn” with a clearly defined gaze action. This reliably triggers the subject’s response. Right: The robot uses gaze to indicate to the human where to place the object. Again, this reliably elicits the correct response.

We also tested another condition, which show the importance of beliefs attributed to the robot. After a training phase where the subject has to interact a few times in “speech + gaze” condition, he has a strong representation of the shared plan between him and the robot. During the test phase, we introduced a condition called “solo”. In this case the robot is using only gaze, and rather than using a shared plan, it uses a plan it which it is the actor of all actions. As depicted on Figure 36 the first action is done by the robot (because the subject didn’t choose to act first), therefore the subject, according to the shared plan he has in mind, should do the next action. Indeed in the this case he actually started to move the arm in order to do

his action, but the robot started to move also causing him to cancel his motion and show an incomprehension facial expression.

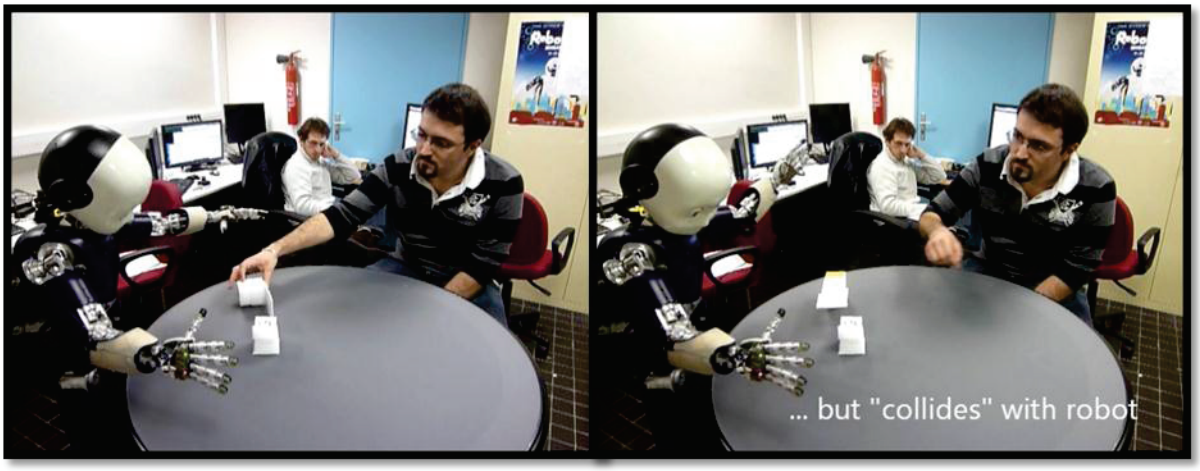


Figure 36 Left: In the no-shared plan condition, the box has tipped over, and the human is placing it in its correct upright position. This indicates that the human is actively ready to cooperate. Right: When his "turn" to move the toy block at the center comes, the human begins to effect that move, but then sees that the robot is to make the same move. This "collision" results in the human withdrawing his initiated movement

Indeed we can extract two ideas from this HRI scenario:

1) The privileged way to express our direct intentionality relies in the gaze and body language. Our gaze is always targeted toward what we are speaking, or thinking about, allowing the observer to also look at it and therefore share our attention. Given the context, gaze can describe an action that we expect to happen next, could it be done by us or by our partner. In this case the observer is able to understand our intentionality and act accordingly. In the solo condition at some point the intentionality of the robot and the subject did not match. The contextual cues of the robot intentionality were not enough discriminating to overcome the shared plan representation that the subject was maintaining, creating the collision of Figure 36. The robot describes other's action by looking at him and at the object he will manipulate, however it describes its own action by looking at the object and then acting. It is quite reasonable to predict that the same kind of collision would happen in the case of a "programed" human / naïve human interaction. Keeping in mind that the shared plan representation gives a form of priming effect about the next action to be executed, a form of prediction of the partner's intentions which we take into account while acting. To come back to the first chapter, the perceived intention of

our partner is a mixture between what we think it should be, and what he actually described using his gaze.

2) The robot, based on his behavior, is attributed intuitively intentionality. There is no evidence that could be investigated about the attribution of goal, however since the subject surely has a goal in the context of a shared plan, he may consider that his goal is shared by the robot. But is it possible to affirm that the subject believes that robot possesses the ability to pursue a goal? A data structure may hold the consequences of a plan, but does it allow saying that it represents a goal? Executing such a plan is just the consequence of choosing a random set of conditions. It would be a good question to ask to the subject afterward. It is the same concerning intentionality: by using gaze the robot expresses the next action he is supposed to do (or wait for), making the subject to attribute him an intention. However, it is likely that many people would be outraged if we were attributing the robot "intentionality".

Addendum : At the time of writing 11 naïve subjects (Figure 37) have been recorded and the data is being analyzed. During this experiment the robot did more than 250 motor actions and the system handled various unpredicted behaviors of the subjects without a single crash.

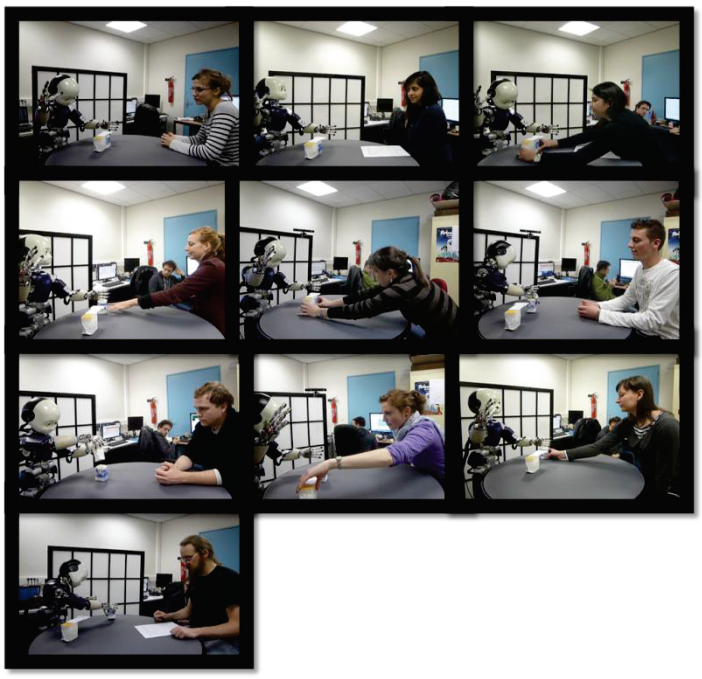


Figure 37: The 10 naive subjects of the last CHRIS experiment.

Discussion

Cooperation is one of the most powerful applications of our cognitive skills, by having our body expressing our mental state we are able to share desires, intentions and emotions. On the opposite side, having our counterparts sharing their mental state the same way enable us to understand will of others. Using those abilities we are able to express and receive concepts including an action or a shared plan. However, this communication process can be subject to noise and only reflects the mental state at a given time. Although we call this plan "shared", it is a fact that each participant owns his own representation of the plan, that he believes the other shares. This phenomenon occurred in the experiment presented; at some point the robot mental representation of the shared plan was not the same as the subject. Communication is a two edged blade: it allows transmitting mental states to others by physical interaction; however it can be misunderstood therefore creating incoherencies in our understanding of others. In the next chapter I will describe how two robots could really share the same plan, the exact same mental construct, but it is something impossible to achieve with two humans. However, robots will have to interact and to cooperate with humans so it is required that they are able to communicate concepts using the "human way". By building a robot and testing how subjects understand his goals and intentions according to his physical behavior, it will give us important insights about what the robot should analyze on the human body in order gain empathy and understand his needs. At the time of writing, the human was taken into account only based on the physical changes happening in objects on the table: his gaze was not monitored, his face was not analyzed, his hands were not tracked. Therefore, the robot had only very few cues about the human status which disabled any possibility of a more "aware" interaction. Since communication seems to be so rooted in motor expression, it is a requirement that a communicating robot is able to understand and express ideas through his body as well as his speech. This is a really strong argument in favor of humanoid robots, especially those with a human-like face. A robot having a mouth, eyes and arms used in a meaningful way will be much more likely to be attributed to goals and intentionality than a "wheeled platform with a screen". And as I mentioned at the early beginning of this thesis, while it will be progressively difficult to be sure if the robot really possess a goal, intentionality or a consciousness, we can at least make it act as if this was the case.

Chapter IV

Abstract Cognitive Machine(s)

Introduction

Robot cognition is a whole new area of research. New robots are popping out every month around the world, each one of them having his own body, his own particular features, his own software, to name a few. Although the situation is evolving very fast, there is still a huge diversity in the software used worldwide to implement robot cognition, and many parts of those implementations are redundant. Usage of a middleware can partially cope with this problem while shifting it to another level: multiple middleware exists, they are not written to be compatible and the choice of one of them is a crucial point in robotic architecture. Moreover, at a "higher scale", software is not the only major issue that robotic research is facing: heterogeneity is also appearing within the different architectures implementing robot cognition.

We have seen how humans were able to cooperate in order to achieve a shared physical goal, to teach or to learn. The concept of communication is central to all those activities; in order to share his intentions an agent has to be able to communicate with others. I would argue that the communication methods used by an animal species derive directly from its sensory apparatus. As human, our main communicative senses are:

- Audition, we developed spoken language,
- Vision, we developed body language, writing, sign language,
- Tactile sensing, we developed brail writing.

Of course the sensory apparatus is not the only requirement for a communication mean to emerge: the body should possess an organ able to produce a stimulus perceptible by those sensors. Spoken and sign languages are the most convenient: our body can both create a stimulus (sound or visual posture) and perceive it as the other agents around. The writing case is a bit special since it uses a physical artifact to transmit a message at a higher distance than the simple range of the voice. Other animal species use other sensory modalities to communicate and transmit knowledge: fishes use electricity (Hopkins 1974), insects use pheromones (Wyatt 2003) and chemical communication is spread among the animal kingdom (Taga and Bassler 2003; Wyatt 2003). But despite the means used, the process of communication is always transmitting an idea from one individual to another, which means give a form to the contents of the mind so that it can be transmitted as a physical signal,

send it to the receiver who will have to perceive it and reinterpret it as an idea. Today's technology (wifi, bluetooth, more generally wireless technologies) allows transmitting any kind of digital data between two remote devices in an imperceptible and quasi instantaneous manner, clearly more efficient than any of our naturally evolved means. The end point users of these devices are humans, who still have to use their own sensory apparatus to format and read the messages transmitted over the network. It means that for any thought we would like to share, we have first to express it in some media (it could be text, sound, image or video); then we should transform this message to a digital format before sending it. On the other side, the recipient will have to do the reverse process: after the raw data has been played by the device, the message has still to be sensed and understood. Indeed we can see our sensory apparatus as a modem (Modulation-Demodulation) interfacing our minds with the physical world. The contents of "minds" of a robot will always be some kind of data structure stored within a digital memory, therefore if a robot has to transmit any kind of idea to another, it could just send this "data structure" over the network, allowing an instant and noiseless communication, like the old human dream of telepathy. However, this will be possible only if the two robots share the same data structure, which leads to the problem I mentioned before: robots are facing the heterogeneity of their cognitive architectures.

Thus robots can only share knowledge between them using the "natural mans". When watching Star Wars, I always wondered why C3PO and R2D2 were speaking together, why did they have to talk? Maybe did the robotic revolution of the Star Wars world miss the opportunity to invent a platform independent cognitive architecture? Fortunately in our world a few projects, including CHRIS (Cooperative Human Robot Interactive System) which supported this thesis, are trying to cope with this problem at various scales. The impact and importance of such a unified robot cognitive architecture is beyond the scope of imagination. A shared knowledge representation would produce an explosion of robot learning: by allowing robots to learn instantly from the experiences of others despite their geographical location, the teaching curve of "newborn" robots will be inexistent. Of course such a shared experience framework is impossible to achieve with no human in the loop, and many technical issues will have to be identified and solved by human operation. In this chapter I will identify some of the initial problems identified while working on the merging of

experience and propose solutions or ideas to cope with them. In this chapter, I'll focus on the technical feasibility of what I named the Shared Experience Framework, while I'll keep all the various philosophical discussions raised for the later discussion.

Various scales of heterogeneity

Robotics is facing a significant of problems, some of them are inevitable such as hardware limits, perception and manipulation issues, however others are created by us and the current technical situation. Among them is the incredible diversity and lack of homogeneity in all domains of robotic research, from the hardware used to build the robot to the cognitive architecture implementing its "brain", passing by the middleware linking both of them. In this chapter I will review the heterogeneity problem at each level, and point at the solutions that are rising to face it.

Robots hardware heterogeneity

When one thinks about the differences between two robots, the first thing noticed is of course the body. When it comes to building robots, the shape of the machine is limited only by the imagination, and certain physical constraints. Most of the bodies are inspired by nature and mimic various living beings like dogs, spiders or humans. However, there is also the possibility to create other "things" like wheeled platforms carrying a tactile table, or swarm of miniature robots. Obviously, one major issue concerning this body shape is that not all of the robots can act in the same environment, and that all the actions described cannot be implemented in the same way. There is nothing that we can change at this level: although a humanoid and a dog can both grasp a ball, each of them will need a specific implementation of the action grasp. The bodies' heterogeneity is a problem we will have to face, although the use of motor primitives with a robot specific implementation is already a partial solution ((Lallée, Lemaignan et al. 2010; Lallée, Lemaignan et al. 2011), attached as appendixes 1&2).

However, apart from the global body shape, the hardware heterogeneity can be a problem of another kind. During my PhD I worked extensively with the Italian Institute of Technology on the iCub robot. We had an older version of the robot in our lab in Lyon, while they had a brand new version which was using a more advanced control method called torque sensing. We had this software module implementing the motor primitives that I described before, but it turned out that a grasp using torque sensing and an "old style" grasp were different enough to require two different implementations. Fortunately the calls to the specific implementation was platform independent, however this example shows that an upgrade of the robot hardware changed it, from a software point of view, into a different robot. Even

more dramatic, within the CHRIS project, the architecture we developed was required to run on truly different robots including iCub and the humanoid robot Bert, although the preliminary definition of motor primitive interfaces allowed us to control and exchange plans seamlessly between the two platforms.

Robots Software heterogeneity

The number of new "cognitive" robots is increasing each year, both as commercial products and research platforms. Each robot possesses its own software architecture, thus resulting in a huge loss of time and money. I will describe in details the implications of this problem, and the various software architectures engineered to cope with it.

Problem

When one starts to work in cognitive robotics, like starting a PhD, he is often introduced to the specific software employed by his laboratory or on the robot that he will use. Most of the time the algorithms developed to handle various aspects of the robot cognition are more or less independent from the platform used; for example the Self Localization And Mapping (SLAM) algorithm used for navigation of mobile robots does not change from one robot to the other if we consider that the motor commands are implemented in other robot controllers which are just called by the algorithm. However, if one has implemented such an algorithm on one robot, and he wants to use it on another robot, he will then need to re-write most part of his code so that it will fit the new robot software architecture and be able to access the new robot sensors and motors. Developing on two different robots is like developing on PC and Macintosh: even if the programming language is the same and the algorithms are identical, the ways to access the platforms are so different that nearly all the code needs to be rewritten to achieve the same result on both. PC and Mac are the two main platforms available on the market, making this effort acceptable, however in the case of robots the number of different platforms increases each year. This is the reason making many developers, researchers and students in different laboratories to solve the same problems, using the same algorithms but at the same time preventing them to efficiently share the results of their work. On computers side the hardware is very different and the same kind of problem occurred at some point, however abstraction layers have been implemented to allow the developers to write software without caring about the hardware which will run their code. Multimedia abstraction layers like DirectX or OpenGL became a

requirement because of the diversity of hardware which spread in homes. The "boom of domestic robots" has not occurred yet and robots are still confined to laboratories, since researchers like to reinvent the wheel the problem is not really one at the moment, however it will be as soon as robots will become commercial products.

The existing solutions

Many solutions are emerging to cope with this problem of software heterogeneity; they take the form of collections of software libraries and programs built with the idea of handling robotic development. Each of them has pros and cons and choosing one is a matter of technical integration (most of these environments have been designed for a specific robot, even if they can handle in theory any kind of robot), of taste and of politics (sometime a project or a laboratory will choose one as the standard platform). I won't enter the details of every solution here, good surveys are available (Biggs and MacDonald 2003; Kramer and Scheutz 2007) and the main systems are presented through academic publications :

- URBI (Baillie 2005)
- ROS (Quigley, Gerkey et al. 2009)
- Player/Stage (Gerkey, Vaughan et al. 2003)
- YARP (Fitzpatrick, Metta et al. 2007)

Cognitive Architecture heterogeneity

During the course of this PhD I worked within the CHRIS project, which led to several papers on aspects of cooperative human-robot interaction (Lallée, Metta et al. 2009; Lallée, Warneken et al. 2009; Lallée, Lemaignan et al. 2010; Lallée, Madden et al. 2010; Lallée, Lemaignan et al. 2011), some of them being attached to this thesis as appendixes, and a quite large collection of software modules. When the project started, we had a clear view of one scenario we wanted to achieve which was prototypical of the interactions we were interested in: a human and a robot, cooperating toward the goal of building a small Ikea-like table ((Lallée, Lemaignan et al. 2010; Lallée, Lemaignan et al. 2011) attached as appendixes 1&2). In the end we managed to achieve in reality what we had initially hand coded at the project outset (our initial wizard of Oz HRI). Because of the nature of the project, from a software point of view, the design of the architecture progressed concurrently (or rather, step-wise) with its implementation. This caused an intensive use of modularity (division of

the architecture in independent software modules communicating over YARP (Fitzpatrick, Metta et al. 2007)). From the software engineering point of view I learned during this PhD that modularity is a two-edged blade: it helps to maintain on going and long term developments by really separating independent modules and allows replacing / recoding one of them and maintaining interfaces. However, sometimes the design of the architecture is done sub-optimally. This can be because we do things to solve a specific case instead of a more generic one, or because we do not catch the "global picture", or even for political reasons. In the end it can result in the division in independent modules of a function that could benefit from being integrated. The CHRIS architecture in the end of the project is represented in Figure 38, in this figure each box represents an independent software module and all the communication between modules is done through exchange of messages on the network using YARP. I will not explain the architecture in detail here (for reference (Lallée, Lemaignan et al. 2011) attached as appendix 2). However I would like to emphasis that while it allowed us to successfully cope with all the aspects covered in this thesis, the architecture has been inherited and improved.

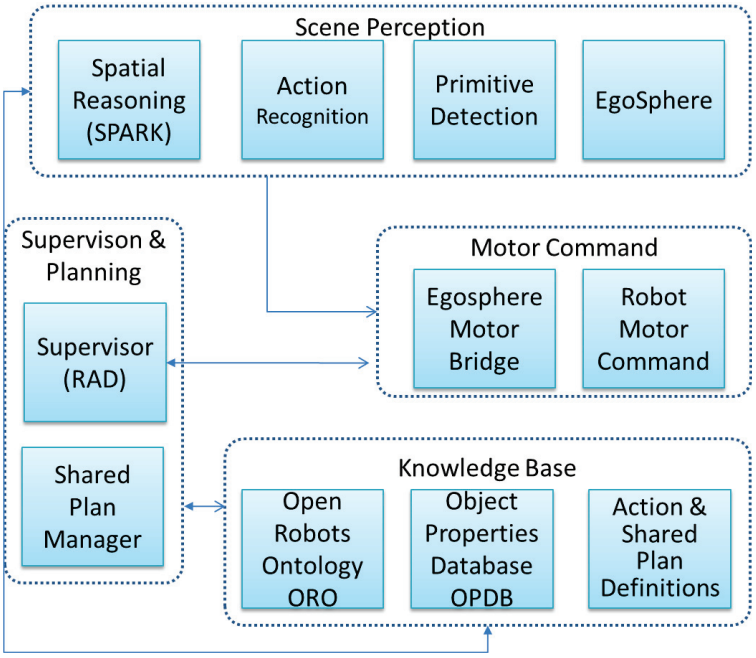


Figure 38: CHRIS architecture in the last year of the project (2011)

At first, the "Knowledge Base" idea seemed nice, it was designed to store semantic and stable properties for objects and the various action definitions in independent modules. With time we realized that every objects properties and relations between objects should be

stored together in a compact and accessible way, thus merging the Egosphere and the knowledge base. Moreover, there was no reason to keep the definitions of actions and plans separated from their manager module. We could have tried to change the architecture; however it would have required a huge transformation of many modules and created several inconsistencies. The best choice at this point was to use concepts and a few modules from CHRIS and wrap them in a freshly brewed architecture for a new project called EFAA (EU FP7 Experimental Functional Android Assistant). Built on the experience of CHRIS, the EFAA architecture (Figure 39) can cope with the same range of tasks, however further developments and maintenance are made easier by a smarter design and a smaller number of modules. The major improvement is the implementation of the Egosphere by a module called Object Properties Collector (OPC): where in CHRIS only spatial information was stored, EFAA extends the concept by storing all properties about objects (spatial, semantic, relations, affordances, etc.). Therefore all other software modules can update and access them in real time. Although there is no doubt that the EFAA architecture is an evolved version of the CHRIS one, we encountered a well know problem in software development: backward compatibility. Data structure formats and communication protocols have changed ending in an impossibility to use the old knowledge base: action definitions, plans and knowledge about objects is impossible to transfer from one version of the architecture to another. While in this case it is not really a problem since in a sense CHRIS serves as a point of departure for EFAA, I fear that such inconsistencies between different cognitive architectures will become an issue. At the time of writing there is no consensus or attempt of formalization of a standard knowledge base format. I mentioned in the introduction of this chapter that robots should be able to exchange or share knowledge in real time bypassing human like communication means. The first step toward this direction is to define such formalism. It is not my goal to do so here, it is a daunting task that should be the outcome of an international project involving the major actors of robotics today. However, such a project will take years to deliver a usable format while we should start to think about the implications and requirements of a shared experience framework. In the following part of this chapter I will describe another cognitive architecture, which I designed independently, based on the knowledge I acquired during CHRIS. The core principle of CHRIS and EFAA is to abstract the cognition of the robot from the perceptive and motor layers. In doing so, the cognitive machine becomes robot independent and the same software/knowledge can be

used on different robots without any change. However the sharing experience and perceptions over a community of robots requires more than “platform independent software”, it needs something like a server to make the sharing of experience among the robots transparent and automatic. The architecture that I will present is called Central Cognition; it is a standalone software which embeds the knowledge base implementation (Egosphere, actions, shared plans) within a multi-robot control system. Within this framework, all the experiments done by individuals are shared by the whole system in real time.

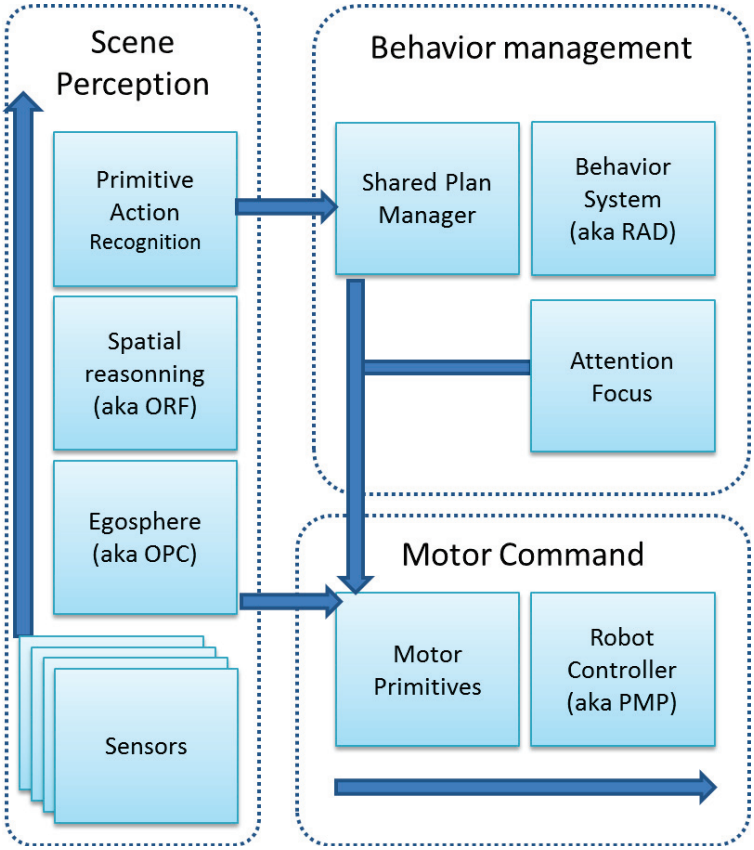


Figure 39: Status of the EFAA architecture at the time of writing.

Shared experience framework

Central Cognition

Central Cognition is a software system designed to maintain a shared knowledge base between multiple robots, while at the same time supervising the behavior of those individuals (Figure 40). On the perceptual side it implements a list of perceptions, which are interfaces to sensor dependent APIs (application program interfaces). At the moment, visual object recognition is provided using iCub and Nao cameras, while agent recognition is provided by the Kinect API (Kinect is an inexpensive rgb/depth camera). More interestingly, it also reads from the OPC (object property collector), which is the EFAA Egosphere. Indeed since they use a similar format for storing properties (see Appendix 4, the EFAA's OPC Specification), EFAA Egosphere can update central cognition Egosphere in a kind of "composite perception". As a concrete example, the OPC is updated by a tactile table and an environment initialization file; therefore, central cognition will also perceive the information coming from those sources. This mechanism is quite important, since all robots in the world cannot (should not?) be controlled by a single software, multiple "control points" should be able to share information among them. Indeed the Egosphere concept can be extended hierarchically if we assume that an Egosphere (a list of properties for objects) can serve as the perception of another one. As an example, the Egosphere built by the sensors of a platform in Italy and a another one in France could contribute to a European merged Egosphere, therefore allowing French robots to "see" what's going on in the Italian lab. In a more domestic application, let's assume that you are looking for your passport. You ask your robot assistant in the kitchen and it will tell you in that your passport is seen by your robot dog on the living room table. While this is already a nice feature, it could be nice to have the dog to bring the passport. Two remarks on this point: first, the perceptions are shared. This means that a command heard by one robot is also heard by others, therefore I can ask to the dog directly as if it was next to me. The second note is about how the control of robots is done, it is achieved through what I called Local Cognitions.

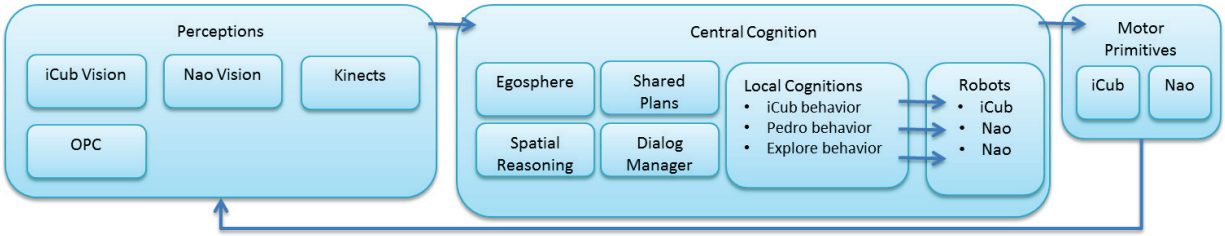


Figure 40: Central Cognition architecture at the time of writing. The use of the OPC from EFAA as a perception and of the PMP for iCub implementation provides a step toward backward compatibility.

While Central Cognition is the entity storing all the shared knowledge base, it also keeps track of all robots. For the system a robot is the implementation of motor primitives' interfaces (including text to speech). Every robot is associated to a Local Cognition, which is an update loop defining its behavior: it is responsible for taking a look at the Egosphere, listen to possible human commands and reacting appropriately. I decided to lock the control of the robot at the local cognition level because this allows an easy customization and definition of "characters" which are important for acceptation of machines as friends. Indeed the "speech style" is local: iCub could say "*Good Evening Stéphane*" while our Mexican Nao (Pedro) could say "*Hola buddy!*". However a behavior is not restricted to speech, indeed it defines all the interactions that the robot will have with its environment. For example I implemented a very basic behavior that does not involve any human robot interaction and just make the robot to focus his attention on the various objects present in his proximal environment. A more evolved cognition could be a needs-driven one like it is the case for the artificial intelligence of the Sims (*www.**thesims**3.com)* or the life simulation Creatures (Grand and Cliff 1998): in those models the agents possess a list of needs (hunger, entertainment, social, etc.) and each action has an impact on those needs, therefore allowing the agent to choose the best action in order to satisfy its needs. I won't give any more insight on this model, although it is probably the most promising system for an autonomous agent; what I want to emphasis here is that the behavior is what makes your robot "unique". Although the same Local Cognition can be attributed to multiple robots (they will have the same behavior), a unique Local Cognition can be handmade for each robot, making them individuals rather than a single omniscient entity. It will likely be easier to interact with and socially accept robots if they are perceived as individuals. We will return to this notion, but quite intuitively we can say that people strongly bind one mind to one body, since all biological organisms know are done that way. Moreover, having this kind of

distributed control allows an easy maintenance of the robots pool available to Central Cognition. Indeed Central Cognition (CC) is a “computer voice”, a virtual agent in some sort, which you can interact with as with any other robot, although CC has “administrative privileges” over the other robots.

By interacting with CC you can ask to “wake up” or “put to sleep” any of the robots controlled, you can also access the Knowledge Base, ask for the content of the Egosphere or the definition of plans, etc. Indeed CC, despite its centralizing role and its ability to manage the state of every robot, is not the system managing the behavior of every single robot. The true behavioral choices are done within each Local Cognition, while the possibility for Central Cognition to directly require the execution of a plan or a primitive action on a specific robot is left open. Indeed at initialization CC delegates the behavior handling of each robot to a Local Cognition (it can change the associated cognition at runtime), however it has access to all the capabilities of the pool of robots, to the Egosphere and the whole Knowledge Base, it can be therefore considered as another “robot” whose body will be a distributed system of sensors and multiples robots. I will discuss more in detail the problem “One mind over several bodies” in the last chapter, for now I will just present several technical and concrete aspects of the knowledge sharing process.

Experience Sharing

Throughout the thesis I have described various processes of learning for the robots. Robots can learn from interacting with their environment on their own, from the formal teaching of a human or by observation. All the knowledge gathered by this learning is formatted and stored in the shared memory of CC, no matter who created it and how, every individual robot who is part of CC can access it and use it, likewise for the real time perceptions of others robots. Such a merging/sharing of information is not a trivial task, indeed very little research has been conducted on this topic.

At the perception level the problem of sensor fusion appears. Multiple sensors can perceive environments which overlap; therefore the same object can be detected by multiple sources, while in the Egosphere only one representation for this object has to be stored. I solved this problem by having the Egosphere responsible for its own updates: it is attached to a list of perceptions, which are polled for perceived objects at a given rate. If the same object is

being perceived by multiple perceptions then a mean of the contradictory information is calculated while the properties are concatenated. Again we face the problems described in chapter 1 about fusing information in a hierarchical bottom-up flow of integration. While those are quite hard in the case of extracting symbols from raw sensory information, when dealing with perceptions that are already producing symbols the problem is much easier. Indeed individual objects are assigned a unique name; therefore it is easy to match multiple instances of the same object perceived on several sources and to merge this information. Of course several synchronization mechanisms are in place to avoid robots or other software threads to read from the Egosphere during the update loop, but apart from these critical time windows, the sum of information coming from all the sensors is available at any time by every piece of software linked to CC. The question of the sharing the results of learning is a bit more novel and rise interesting issues.

Assume that I teach the action “swap A and B” to the robot. I will go through a quite painful process of describing the new composite action *swap* in terms of sub actions and primitives, or I will demonstrate it. Both ways will end in having a compact action definition created, that the robot will commit to the shared knowledge base so it can be used by other robots. Let’s assume that we are working with a large community of users and robots, there is quite a risk that other people will try to teach the action swap either at the same time as me, or before me but in a fashion that I do agree with. Those problems are well known in every software development team and are at the origins of the need for versioning systems. When two or more people collaborate on a shared document, they have to synchronize their work either by having a shared plan (Chapter 3) or by merging their modifications together afterward. Those two methods are not very comfortable for human beings and clearly unusable for the purpose of the Shared Experience Framework. This issue has arisen during the CHRIS project where the knowledge base was mainly stored in text files and the synchronization over multiple laboratories was achieved using the merging functions of SVN (Collins-Sussman, Fitzpatrick et al. 2004). In CC there has been no need yet for this kind of mechanism since the system is a standalone piece of software which is not handling the commitments of multiple users (that is, I was keeping the system in a coherent state and didn’t produce any conflict on purpose). Future work will probably see the appearance of such problems and at some point a merging mechanism for knowledge bases will be

required, as the tools that are being developed in the ontologies communities (Noy and Musen 2004; Völkel and Groza 2006). Our partners in the CHRIS project developed a module dedicated to ontology management, ORO. Open Robot Ontology (Lemaignan, Ros et al. 2010) can store concepts and link them to store knowledge and reason about it. Such an ontology capability could likely be an effective approach to storage of knowledge in the case of cognitive robotics, particularly with the merging processes mentioned above. With the ability to merge multiple knowledge bases, a collective memory of every robot could emerge allowing an omniscient robot swarm (Waibel, Beetz et al. 2011).

Discussion

I presented in this chapter the foundations of what could be the future of robotics. It is interesting to note that at this point I'm becoming unable to find any scholarly references about the topic I'm developing: all the questions revealed by such a central cognitive architecture come from science fiction books and movies. Even within this literature, only very few deal with the topic of a centralized memory. Most of the time it's just to mention this "exponential learning", described by the T-800 in Terminator when speaking about Skynet, which turns the system into an evil artificial genius who decides to destroy humanity. A more interesting case is described in *Ghost In The Shell: Stand Alone Complex* (based on (Oshii, Shirow et al. 2004)) and invites us to follow the evolution of intelligent tanks. All the tanks act as friends to the human characters, they are given orders by speech, are fully independent and communicate together in wireless chat but still being individuals during the day. Each tank takes its own decisions, has its own perceptions and collects its own memories. However at night, all the tanks' memories are "synchronized", therefore allowing each individual to gain the knowledge gained by others during the day. It is hard to not think about a "shared dreaming process". The main interrogation about this in the story is the question of individuality: is each tank a standalone individual? Are they all part of a complex system? After the synchronization process, we can say that there is only one memory which is shared by all the tanks, therefore there producing a single complex individual; however as soon as they start to perceive again on their own they become different beings. At some point in the story all the tanks are physically destroyed and it turns out that their "brain", their central cognition is located in a satellite which allows them to continue to live in the network. They have no body anymore, however their cognition is still there, providing an ability to reason and speak. At some point they even gain a "new body" by hacking the satellite control system in order to be able to move it. In this case robotic bodies can be seen as a way to harvest knowledge about the world but the mind could be an independent cognitive process which does not require a body to exist (apart from the computational substrate of course, which could be neurons or silicon). The development of a collective cognition opens various reconsiderations of the body concept and its relation with the mind. It seems logical that the relation one body / one mind is not true in the case of artificial life: multiple bodies can be associated to multiple minds, with all the possible combinations

implied. One mind can control multiple bodies (which can therefore be seen as a single distributed body) and one body can be controlled or contribute to multiple minds, etc.

All throughout this thesis, I never doubted that a mind can emerge on a machine, although it is a strong position, I would like in the final discussion to speak about this idea and why people say “No, a machine will never be intelligent nor conscious”.

Discussion

The domain of abstract artificial cognitive machines targeted toward human-robot interaction is only babbling yet and produced solutions are research products more than an engineered, ready to use, systems. In the world's largest conferences of field, like IROS, most of the works presented deal with very specialized aspect of human robot interaction. Indeed only very few groups pursue the goal of integrating capabilities ranging from precise motor control to action recognition or spoken language understanding and programming. The major contribution of the CHRIS project (and subsequently of this thesis) is to show how all those standalone capabilities can be merged together in an integrated cognitive architecture, and how such an architecture can lead to real world application. Developing, maintaining and even using such architecture requires a huge amount of work including mostly system engineering and human-human cooperation but it is necessary in order for robotics to take off and stop demonstrating capabilities on "toy cases scenarios". Cognitive architecture design is quite appealing from a theoretical point of view, it allows to draw nice diagrams with boxes related to psychological or neurological concepts (long term memory, dopaminergic system, etc.), however despite their undeniable interest as high level models of the though those systems are most of the time far of being usable in a technical point of view. Surveys of such system are available (Chong, Tan et al. 2007; Vernon, Metta et al. 2007; Langley, Laird et al. 2009) and they provide ways to benchmark other candidates. However it is important to keep in mind that a nice cognitive design and abilities that it should grant to the system in theory is often very far from the real world application: many systems described by elegant papers are just not able to produce any kind of live demonstration. Cognitive architectures applied to the control of a real robot start to appear but are still not numerous (Scassellati 1999; Benjamin, Lyons et al. 2004; Cassimatis, Trafton et al. 2004; Burghart, Mikut et al. 2005; Vernon, Metta et al. 2007; Vernon, Metta et al. 2007) and most of the time they are not available as open source engines that everyone can use (or even if they are, they are so complex and undocumented that it is impossible to adapt them to another platform). This limitation comes mainly from the fact that such architectures are designed to handle one specific robot; therefore they are not thought to be used by other laboratories on other robots. When it comes to the abstract cognitive

machines, those that are truly independent of the robot platform used, the state of the art is even more reduced. To my knowledge the only project similar to what we achieved in CHRIS is RoboEarth which focus on the development of a standard language for robot to store and exchange knowledge on a cloud-based server (Tenorth, Perzylo et al. ; Zweigle, van de Molengraft et al. 2009; Guizzo 2011; Waibel, Beetz et al. 2011). Our approaches are indeed very similar and both projects hold the ambition of managing robot knowledge in hardware abstracted way, allowing multiple robots being to use and contribute to a common memory. Moreover their system seems to be robust, able to produce demonstrations and includes a cloud (web) component that our actual implementation is lacking. The specify of our architecture is its unique ability to learn, execute and edit plans that are shared by multiple individuals and which have been demonstrated to be an essential component of cooperation in humans (Tomasello, Carpenter et al. 2005; Warneken, Chen et al. 2006; Dominey and Warneken 2009; Tomasello 2009), this ability is grounded within a spoken interaction framework inherited from spoken language programming (Dominey, Alvarez et al. 2005; Dominey, Mallet et al. 2007; Lallee, Yoshida et al. 2010) which allow to easily produce new knowledge. Indeed it would be a great achievement to interface our system with the RoboEarth API so that both can benefit from each other's specificities, but as every integration activity such a project would require time, cooperative peoples and cooperative robots.

Perspectives and Inquiries

This thesis aimed to cover all the requirements for developing the cognition of a robot companion, starting from the lowest level perception-motor loop and cortico-inspired associative maps in order to classify the raw information from sensors into concepts and symbols. We have then seen how such symbols could be assembled into an action definition and then how those actions could be used in a cooperation landscape through shared plans, imitation, learning from observation, etc. The last chapter dealt with problems and possibilities introduced by the development of an artificial cognition: the dissociation between the mind and the body. However the original title of this thesis was *"Toward a distributed, embodied and computational theory of mind*", which is a bit inconsistent with my previous statement that mind and body may be independents. Indeed I'm facing a paradox here: I generally agree with the theory that cognition is partially situated, embodied, that our mind is shaped by our body and that our body is controlled in a mind dependent way (Wilson 2002; Anderson 2003). Many psychophysics and neurophysiological experiments account for the theory of embodied cognition, indeed my multimodal convergence model and my Annex 1: A Theory of Mirror Development also suggests that this framework is the right one to explain biological cognition. Humans and other animal species which possess a cortex are developing their behavior through their interaction with the world. Assuming that we can call "mind" the dynamics of the brain processes, then the mind is shaped by the body in the sense that all interactions with the world (perception and action) occurs through the physical envelope and are linked by the brain. The mind is for a large part the knowledge of all regularities extracted from this perception of our universe's rules. Early on it extracts physic rules, it learns that a part of the physical world is an entity that can be moved by sending signals to muscles and that this entity is continuously sending back information about its state within the surrounding environment. It learns that any command sent to the entity will affect the stream of afferent perceptions, and that the command type is directly impacting this change. Moreover at least a part of the mind is directly shaped by the body, which is indeed the substrate (neurons) that makes possible those computations; the sensorimotor organs directly impact the material available for the foundation of the mind.

Solipsism claims that the only thing that we can be sure to exist is our own mind, we cannot have any certitude about the world, about others mind, or even about our body. Although what would be a mind which never commanded and observed a body interacting with the world? Can it be the world or some evil genius ((Descartes and Moriarty 2008), something is sending information to our sensors, and this information is consistent with the orders sent to our motors. In this sense the body is required in order to create the mind, it is the tool used to gather regularities of the world and to build causal relationships. However, once concepts and rules are acquired, is the body still required in order to have the mind running? When a concept representing a world object has been created in mind, one can reason about it without the need of the body. Think about dreams or just "in bed" imagination: one can close his eyes, relax his body, and have his mind feeding itself with mental perceptions. Indeed during sleep the thalamus, which is the gatekeeper between our sensors and our brain, is modulating our perceptions and make them less influent to our mind (Llinás and Ribary 1994; Magnin, Rey et al. 2010). By reducing this impact, it allows the very spirit to take control of what the mind perceives by making its percepts mainly based on feedback. However, this is not inconsistent with the embodied framework: even during these sleeping phases the body is still shaping the mind and the oneiric representation of our self is grounded in the daily perception of our physical envelop. Our dreams include ourselves, objects, and persons and are generally compliant to physical rules. This is easily explainable; by considering the convergence zone principle we see that every single concept or symbol in the brain is indeed related directly or indirectly to a pattern of activity of our sensory layer. In all our life's experience information about our body is present, this entity is an actor of all our memories, and it is in the background of every sensory trace that our brain recorded. As the tool to perceive and act on the world, the body is the core component of the cognition. As I said earlier there is no doubt that biological beings cognition is fully embodied, but has it to be the case in machines?

It is still early to say that the robotic system built along this thesis can be attributed a mind; however we can reasonably say that it is maintaining a sum of its past experience which composes a mental landscape of world grounded memories. This knowledge base could serve as a base for reasoning, and older traces could be evoked as the result of a mental supervisor process. In the following of this discussion I'll call that the "mind" of the

robot: the sum of its memories and the process handling how, why and when it is accessed. My first question about this artificial mind asks about the embodied qualification. In biological being there is no clear separation between the sensory motor apparatus and the so called symbolic level: all symbols manipulated by the brain are derivate of sensory motor representations. An artificial cognition is not bound to these requirements: sensory motor primitive are at some point pure amodal symbols that the brain can reason about. Of course, in order to be perceived or executed, those concepts need to have some link to a body: the visual representation of an object need to be stored in a sensor dependent way (an image pattern for example), and the "grasp" motor command needs to be interfaced by a controller specific to a robotic body. However, if the sensor or the controller changes, then assuming that the new one holds the same interface, then the mind will still be able to perceive and send the same commands. Moreover, even in the absence of sensor or body, the artificial cognition can still manipulate the symbolic layer to produce a mental simulation of an action in order for example to calculate the consequences of a plan. A robot body (collection of sensors and effectors) is required to express the computations of the mind in the physical world and to initially build this mind; however the mind can exist as a standalone computational process, abstracted from the body. I have described the Shared Experience Framework, which allows multiple robot bodies to contribute by their perceptions and knowledge to a centralized knowledge base. However every robot behavior is handled by a separate process, which allows each of them to share knowledge of others and at the same time to use it in its own way and to express its own "mind" by commanding the body it is attached to.

Indeed for humans, the concept of mind seems intuitively linked to the one of individual, which is natural since our reasoning processes are based on the fact that we are individuals. Depending on the mind definition used it can be extended to communities of beings which creates the emergence of a collective mind, which is more than the sum of all the minds composing it. Indeed the idea of a distributed mind is growing and being tested in multiple scientific fields (see (Heylighen, Heath et al. 2004) and (Weick and Roberts 1993) for references on this topic). The collective mind, if it can exist in communities of biological beings, is fairly limited in its possibilities by the fact that two agents need to use conventional communication means. When I mention an "embodied, distributed,

computational theory of mind" I had a view of something deeper than just the processes resulting from the communication of multiple agents. The Shared Experience Framework provides a mental material that is the same for every individual contributing to it, while at the same time allowing each of them to build its own behavior. In this case, where is exactly the mind? All agents possess the same knowledge and only the way they use it is different, therefore could we say that the mind is simply the process of using knowledge to trigger some behavior? It would mean that every robot of the system owns its own mind, but it would also mean that the mind is a process which can be abstracted from the knowledge it works on, therefore creating dissociation between the body, the experience and the mind (see Figure 41, right part). But can we really say that in biological beings the mind is not just created by the sum of all experiences? Is there really a process that is independent of the memory and that handle the behavior, or is every single act we do just the consequence of a memory echo? In the left side of Figure 41 I depicted the mind and the experience as separated, however I did so just to make the parallel with the Shared Experience Framework. I cannot and do not want to give an answer to this question. We reached a point where the implementation of cognition on machines doesn't have to cope with the same requirements as the wet brain. The Shared Experience Framework opens various philosophical questions about the gap between an individual and a community of robots. For example, it is the core of the so called "Stand Alone/Complex" problem introduced at different levels by Arthur Koestler (Koestler 1968) and Masamune Shirow (Shirow, Oshii et al. 1995) (Ghost In The Shell). It will also question heavily the existing philosophies of the mind and the body: a body is required to gather knowledge about the world and to support the creation of a mind, but has the mind to own only one body? Is it transferable from one body to another? Once born, does it need any body to survive? Studying body, mind and consciousness under the engineered light of robotic allows not being dazzled by the sacred light of the Human.

Bernard Werber, in "L'Encyclopedie du Savoir Relatif et Absolu" said with reason that in order to understand a system you have to extract yourself from it. All through this thesis I tried to avoid referring to mind, spirit or consciousness because they are concepts that we experience so intimately that they wield for almost everybody a "sacred" aspect. They are the essence of what we are and it appears to be impossible to reason outside of them. Even when Descartes doubted about everything, he reached the famous statement "I think

therefore I am", therefore considering the thinking process and being conscious of this process as the core of our reality.

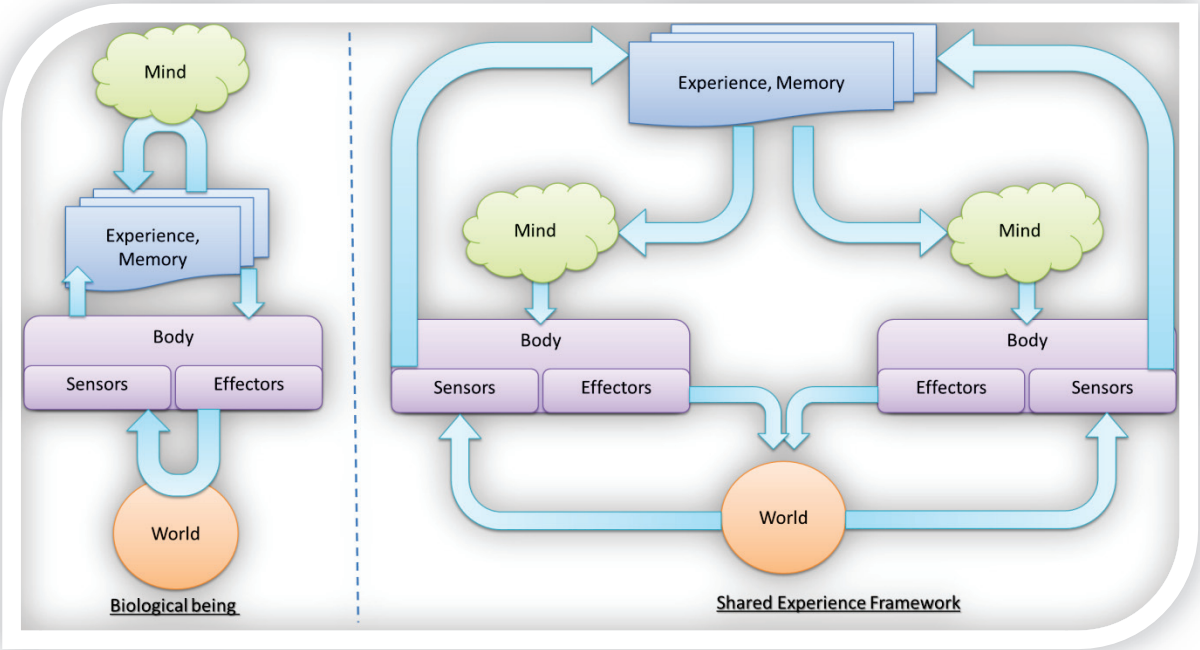


Figure 41: The different interaction between the being components in both living things and in the Shared Experience Framework. Mind have to been understood as the process responsible for taking decisions based on the past experience and the current sensed world.

However mind and consciousness are not elements that we attribute only to our very selves: they are also characteristics of other individuals. We consider that the behavior of others is a direct expression of their mind; therefore if this behavior is something that makes sense to us we assume that the individual responsible for this body owns a mind. If the body is unable to act, as in coma, we have many difficulties to decide if we should attribute spirit or consciousness to the inanimate envelop. Neuroimagery starts to investigate ways to test if the so called consciousness if still present in the brain (Laureys, Boly et al. 2006), indeed corticothalamic disconnections could be responsible of "consciousness loss". Interestingly the thalamus is the interface between the cortex and the body, which could therefore be the link between body and mind. But most people haven't such ways to investigate the consciousness of others, and being honest, we decide to attribute the conscious privilege to someone based on his physical interaction with the world. Therefore what about the case of artificial beings? Could they be considered as owning a mind whenever their behavior will become plausible?

I said in the preface that robotics was an illusion: by improving the behaviors of robots so that they interact with the world as humans do it, we implicitly fool observers to have them to attribute intentionality to a machine. Solipsism states that we cannot be sure about the existence of others and it is indeed true: how can we be sure about the existence of our neighbor's mind apart from observing that his behavior is coherent? The only thing we can be sure about is that some process makes him act as if something was controlling his body in a similar way our mind controls ours. I summarized this idea in Figure 42 where I represented schematically that we are maintaining a model of the self and the other, while the first is the source of our behavior, the second is only a reconstruction based on the observation of other's behavior. Naturally we try to have this behavior to fit a model of a "like us", we assume that others mind is shaped as our own although if we take time to think about it we have no evidence of this. We attribute mind to people based on their behavior, although those controllers could be very much different to the thing we experience as our own mind. Therefore could we have the same argument about our own consciousness? What if this strong feeling that we are some mind-embedded envelop was only the result of ourselves perceiving our own behavior as an appropriate one? When one acts wrongly while being drunk, it is a common afterward justification to say "I was not myself", underlying that our own mind was not controlling our behavior at this time. We observed our acts, but since they were not making sense we do not attribute them as our own decisions. It is a strong assumption, but if consciousness and mind are only the result of one observing that his own behavior matches what he expects, what could forbid a robot to hold them? One could argue that we can still be sure of our mind existence even without acting, by the thinking process, by mental imagery. What is thinking? We can guide it by will, we can force ourselves to think about a specific idea, let say a dog, and we know that we are conscious because the sensory traces composing the dog concept activates in our mental percepts. If we consider the ability to focus our computations on a specific topic as a mental behavior, as our way to act on our direct mind perceptions, therefore the same idea applies: we know that we are thinking because the direction we imposed to our thoughts is the same as the one we perceived. We are able to perceive the effects of our mental actions on our thoughts, on our mental percepts, therefore we assume that we are conscious and that we are behind the commands. So what if the consciousness was just the mind's ability to perceive itself?

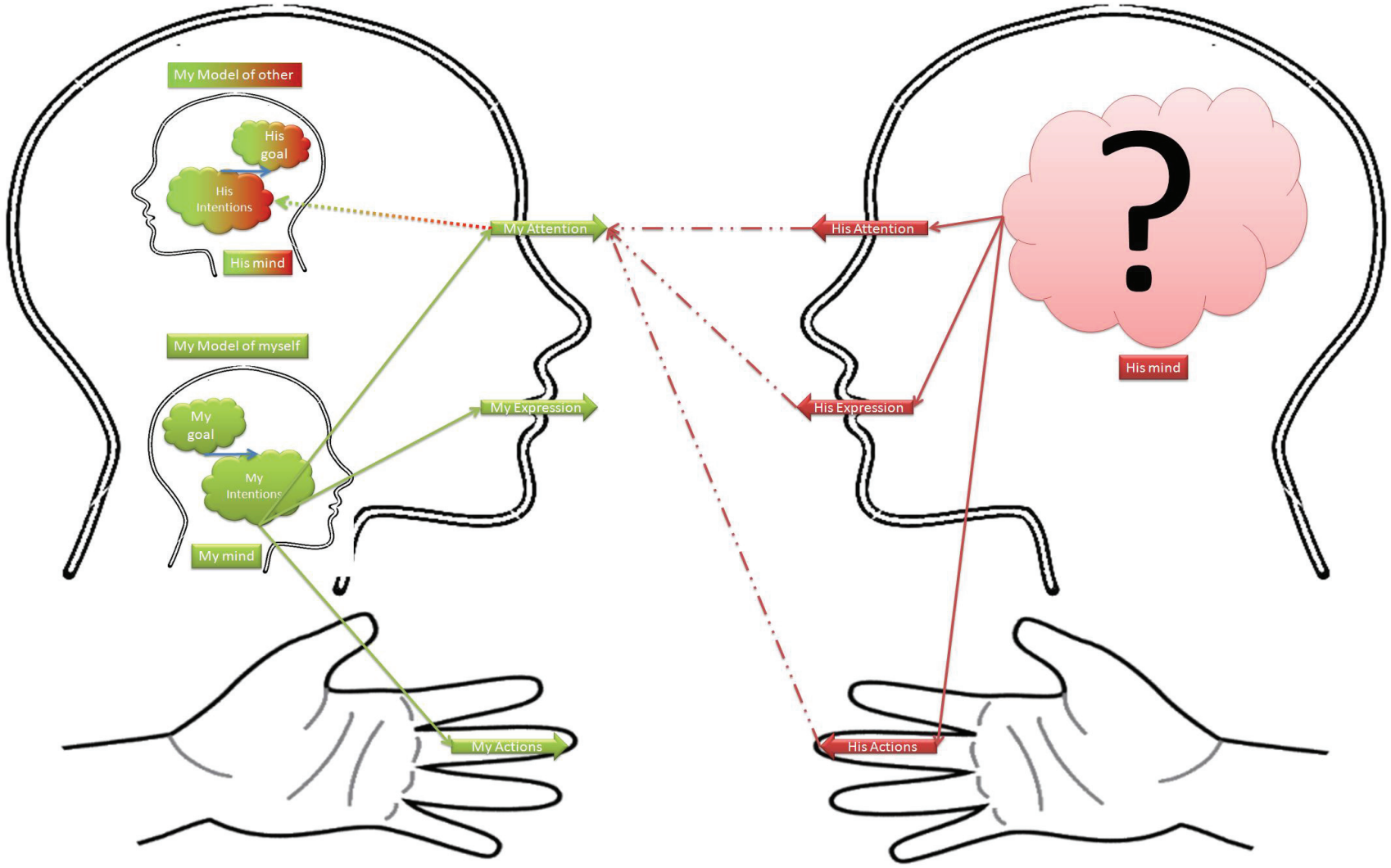


Figure 42Intentionality and goal of others are reconstructed from the observation of their actions and behavior. However, being attributed a mind isn't a necessary condition to prove its existence.

Annex 1: A Theory of Mirror Development

Since Rizzolati discovered neurons that react both to observed and produced actions (Pellegrino, Fadiga et al. 1992; Rizzolatti and Arbib 1998; Rizzolatti and Craighero 2004), the cognitive science community has powerful theoretical object to manipulate: it is a system that could be an explanation for all the primates unique cognitive skills, although ethics, technical feasibility and the general feeling that the explanation for cognition cannot be simple are forbidding any global theory of mirror development to emerge. Indeed many objections have been made to calm down the "mirror neurons excitement" (Dinstein, Thomas et al. 2008; Hickok 2009; Lingnau, Gesierich et al. 2009), producing again a sort of "war of philosophies" which science is fund of. As a "pacifist scientist", in this annex I will not make any claim about the human cognition, I will just describe which mechanisms related to imitation and mirroring could help us to build robots that learn, maybe not as human do but at least in the same conditions as humans.

Learning is a matter of consistency and convergence. One of the most basic properties of neurons is the Hebbian reinforcement, the fact that two cells that are often activated together will strengthen their connection so that they will be more likely to be activated together, etc. Indeed this virtuous circle effect can be witnessed in many aspect of what I call mirror development. Let's assume that a newborn possess a basic untrained motor mirror system, which means that the observation of a body and the self-proprioception activate the same cortical area, but maybe not in consistent ways. However, the expression of the baby will be perceived and, through a controlled echopraxia (unconscious imitation (Buccino, Binkofski et al. 2001)), the caregiver will imitate the baby. However according to my MMCM model, such an imitation would be pulled toward a posture known by the caregiver, therefore having him to express a mixture of the baby's posture and of a posture he is used to take. The inverse phenomenon occurs: the newborn perceives the adult and therefore moves toward this usual posture which acts as an attractor. The loop is described in Figure

43 and has been used as a teaching mechanism to have a MMCM map to learn the mapping between robot encoders and human skeleton recognized by Kinect (Figure 44).

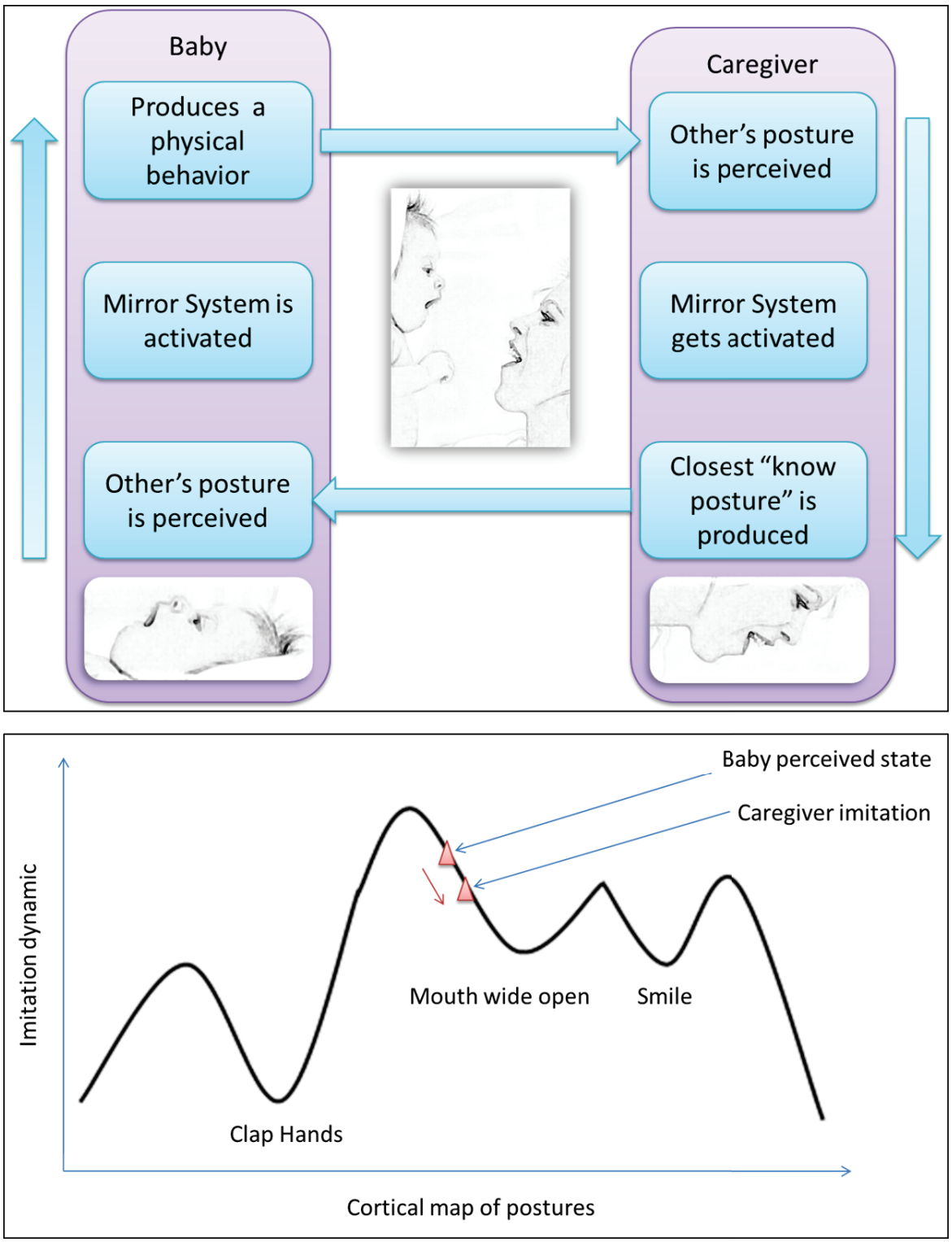


Figure 43: Convergence phenomenon of imitation. Postures known by the caregiver act as attractor and are teached to the baby through imitation.

In this experiment the robot was moving sequentially into different arm postures and the user had to imitate it, producing pairs of kinematic vectors that were taught online to a convergence map. After the map had learned, it could be used to have an imitation of the human by the robot, or to cause a drift in robot motor control based on human perception, as in echopraxia. At the time of writing no further investigation has been achieved and the experiment status is more a "proof of concept" (See video at http://youtu.be/uUquQdnGohE) than a real way to teach useful postures to the robot. However it could be used to test predictions of the MMCM model in against human imitation.

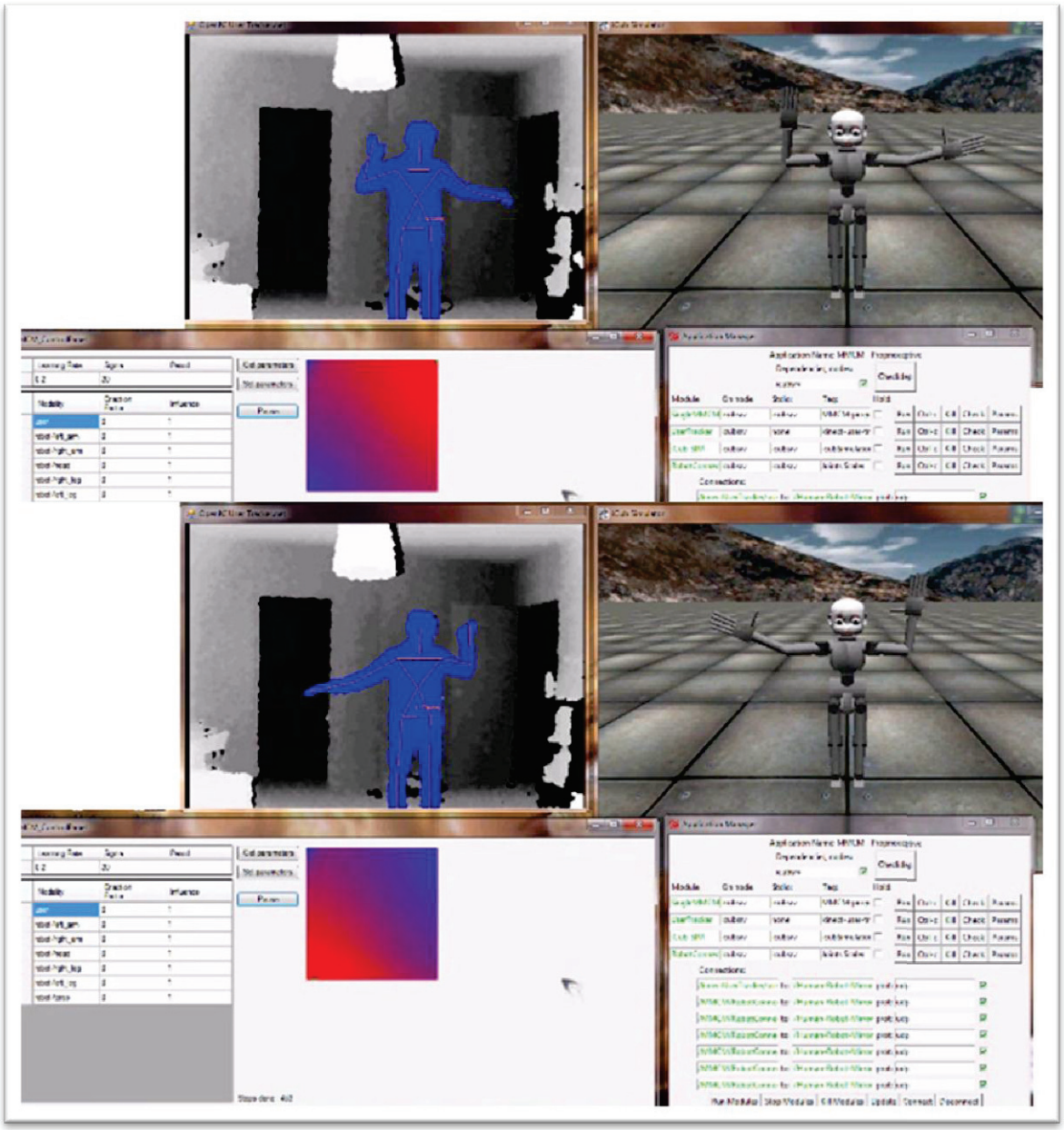


Figure 44: using the mutual imitation cycle to teach a MMCM to solve the correspondence problem between human and robot kinematics.

The same kind of converging imitation is also the basis of the motor theory of speech (for the motor part and how to produce sounds (Liberman and Mattingly 1985)) and of the talking heads experiment (Steels, Kaplan et al. 2002). As for the posture, the spoken name of an object will be driving the imitation loop between the child and the caregiver. For example, whenever the child will say something that sounds at least a bit like “mum” the mother will repeat the correct word and this correct representation both on motor and audio modalities will bring the children a step closer to a good pronunciation.

Another famous example of this phenomenon is imitation of facial expressions (Meltzoff and Moore 1983; Carr, Iacoboni et al. 2003). This two way imitation (child does something, adult imitates, or reversal) is a perfect process to learn behaviors and self to other mapping at the same time. The caregiver is already shaped to produce meaningful postures in appropriate situations, when the child imitates him he shapes himself to produce this kind of useful behaviors. The mirror system could be a very good hypothesis to explain why this tendency to imitate appears, because a simple innate hard wiring of other and self-perception could lead to those mirror development loops and therefore to the creation of the mirror system by synaptic plasticity.

Annex 2: Central Cognition, Implementation Details

Implementation details about the different cognitive architectures described in this manuscript are given in the related papers ((Lallée, Lemaignan et al. 2010; Lallee, Madden et al. 2010; Lallée, Lemaignan et al. 2011) attached as appendixes 1, 2 & 3), however Central Cognition, the software designed for handling the Shared Experience Framework hasn't been extensively detailed yet. In this annex I give a few technical explanations about the software engineering involved in this project.

Central Cognition is a C# program which core functionalities are standalone, they do not make use of any library and are therefore very easy to install, run and maintain. Some low level interfaces (like the specific robots controllers or sensors API) require to link CC against third part library for controlling specific hardware, however the .NET framework allows this integration with a minimum amount of effort. I'll first present the core functionalities of CC and then present how it can be used to handle a group of iCub and Naos.

Core

The main functionality of CC is to provide a way to store knowledge (including objects, actions, plans and semantic relations) in a robot independent way. A few static classes are devoted to this job, the most important one being the Egosphere which is in charge of storing the list of every world object (in the broad sense) known by the robot, including objects that are not present at a given time. The concept is similar to the one of the CHRIS's Egosphere (which was representing objects using an ID and their spatial position and orientation) or to EFAA's OPC, however the way to represent the information is much more optimized and intuitive. The Egosphere by itself is simply a dictionary of WorldObjects indexed on their names, the interesting fact is the datastructure representing a WorldObject: it is actually a class which is a part of an inheritance hierarchy as described in Figure 45. Having those items being real objects within the code allows easy management and interesting linkage of functionalities. For example the grammar management maintains list of vocabulary which is made of WorldObject, therefore when the sentence "iCub take toy" is recognized, the words "iCub" and "toy" provide a direct access to the mental representations of those objects.

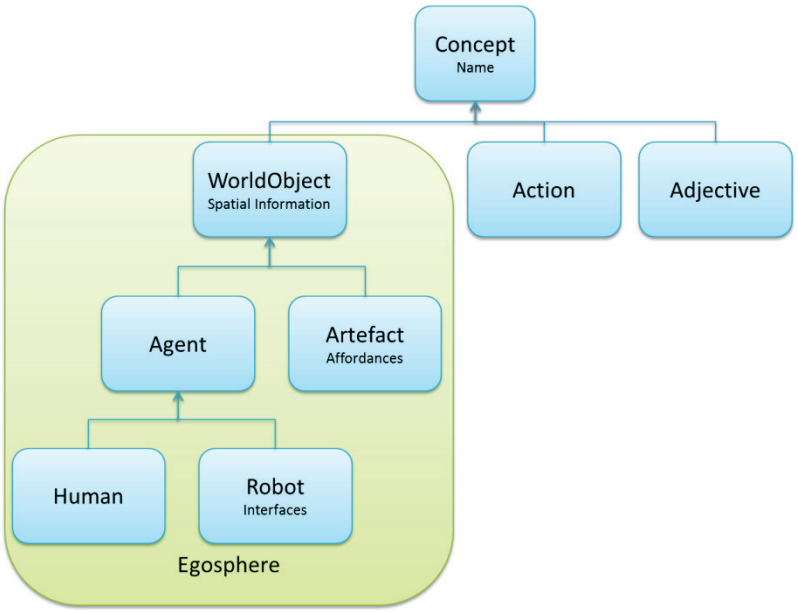


Figure 45: Inheritance among the classes representing every concept known by the robot. The Egosphere is a collection of WorldObject which represent both artefacts (item it is possible to manipulate) and agents (robots and humans).

Three main types of objects coexist now in the Egosphere: Artefact, Human and Robot. An artifact is basically only spatial information (position, orientation, size) and an affordance (a grasping configuration to be used by the robot when grasping those objects). Human are specific entities, but at the moment they do no hold special properties apart the fact of being an agent (therefore contribute to the vocabulary list for generative spoken grammars). Robot is the most interesting class, since it handle both the robot spatial information, but also an interface to be implemented by the final robot type (iCub, Nao, virtual avatar…). A basic motor primitive set needs to be implemented; optionally the robot can implement custom primitives, which allow recording/replaying of motor postures (for example to teach the robot how to wave).

```
abstract BehavioralErrors lookAt(WorldObject where, bool waitActionDone);
abstract BehavioralErrors reachAt(WorldObject where, bool waitActionDone);
abstract BehavioralErrors graspAt(WorldObject where, bool waitActionDone);
abstract BehavioralErrors release(WorldObject what, bool waitActionDone);
abstract BehavioralErrors                    executeCustomPrimitive(string p);
abstract MotorPrimitive.PrimitiveInstance    learnCustomPrimitive(string p);
```

Figure 46 : The functions to be implemented by specific robots. It includes both a basic set of primitives, and the ability to build custom primitives (sequences of postures that are recorder, assigned to a name and replayed on demand).

Speech capabilities are also provided to every robot by providing him with an instance of SpeechRecognizer (Microsoft Speech API) that can be attached to the robot specific microphone, it also provides a standard text to speech, which can be overridden (for example the input of the Say command in the case of iCub is forwarded to a module responsible for moving the lips).

Another main static class for CC is the SharedPlanManager. It maintains a representation of the plans known by CC. A specific shared plan can be retrieved by every robot and then can ask for its execution. The plan is achieved action by action: if the robot is the subject of an action then the according motor primitives are fired; in the case of the action to be accomplished by another agent, then the robot puts itself in waiting mode, waiting for the central ActionRecognition module to detect this action. A re-engagement mechanism is also implemented so that after the while the robot will ask to the user to execute its part. The action recognition at the moment is not implemented in CC, only the robots actions are forced to be recognized when achieved so that robots can synchronize themselves, in a human action case the user is expected to say “done” when the action is accomplished. An effective action recognition based on Kinects input will be the next improvement of CC.

Indeed since multiple sensors (Kinects, robot cameras, tactile table…) can feed the Egosphere, a unifying mechanism was needed. Here again an abstract class, Perception, provide the necessary interface to have any kind of sensor to feed the Egosphere with a single format. Perceptions are basically threads that call periodically an Update function which returns the list of WorldObject perceived. The Egosphere is responsible for handling those signals and managing different perceptions which update the same objects (it does a mean of their different information, therefore granting a more accurate localization).

Extension

As example of how sensors can implement Perception or Robot, CC provides a few ready to use classes.

Perceptions

Spikenet

Spikenet perception is a template matching visual system which tries to recognize objects on a Yarp stream of images. It is based on a commercial system (Spikenet (Thorpe, Guyonneau et al. 2004)) that we have been using along the CHRIS project. Model files can be learnt, loaded or saved at runtime.

OPC

CC is mainly is sequel of the CHRIS project, however it is also compliant with the EFAA project, which uses a module called OPC as an Egosphere. The project related sensors (Reactable & Kinects) have already modules feeding the OPC. Therefore instead of coding a new implementation of that module for CC, it was easier to have a special perception which uses the OPC as the source of its data. OPC_Perception is polling the OPC, retrieving the objects and forwarding them to CC's Egosphere which allows a transparent integration of all the EFAA's sensors (see Figure 47).

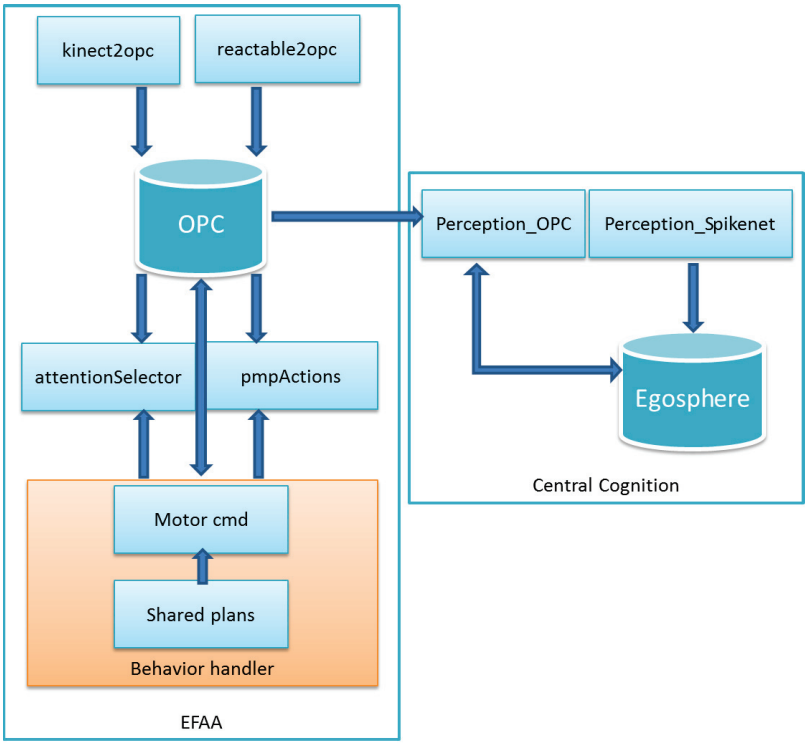


Figure 47: Integration of the EFAA's OPC in CC through the use of a single perception.

Robots

iCub

The main robot tested with CC is the iCub. There are even two different implementations of the robot using different motor controllers (respectively CHRIS and EFAA controllers). They both override the custom primitive learning by allowing the user to put the robot arms in a compliant mode, allowing kinesthetic teaching by recording sequence of joints angles for later replay.

Nao

The Nao's implementation is mainly based on the work done for the Robocup@Home 2011 competition. The motor implementation for basic and custom primitives was directly imported as a .NET library to allow execution of Choregraphe based actions (standard primitives) as well as kinesthetic teach actions.

Appendix 1

Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception

Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception

Stephane Lallée, Séverin Lemaignan, Alexander Lenz, Chris Melhuish, Lorenzo Natale, Sergey Skachek, Tijn van Der Zant, Felix Warneken, Peter Ford Dominey

Abstract— **One of the long term objectives of robotics and artificial cognitive systems is that robots will increasingly be capable of interacting in a cooperative and adaptive manner with their human counterparts in open-ended tasks that can change in real-time. In such situations, an important aspect of the robot behavior will be the ability to acquire new knowledge of the cooperative tasks by observing humans. At least two significant challenges can be identified in this context. The first challenge concerns development of methods to allow the characterization of human actions such that robotic systems can observe and learn new actions, and more complex behaviors made up of those actions. The second challenge is associated with the immense heterogeneity and diversity of robots and their perceptual and motor systems. The associated question is whether the identified methods for action perception can be generalized across the different perceptual systems inherent to distinct robot platforms. The current research addresses these two challenges. We present results from a cooperative human-robot interaction system that has been specifically developed for portability between different humanoid platforms. Within this architecture, the physical details of the perceptual system (e.g. video camera vs IR video with reflecting markers) are encapsulated at the lowest level. Actions are then automatically characterized in terms of perceptual primitives related to *motion*, *contact* and *visibility*. The resulting system is demonstrated to perform robust object and action learning and recognition on two distinct robotic platforms. Perhaps most interestingly, we demonstrate that knowledge acquired about action recognition with one robot can be directly imported and successfully used on a second distinct robot platform for action recognition. This will have interesting implications for the accumulation of shared knowledge between distinct heterogeneous robotic systems.**

Manuscript received March 10, 2010. This work was fully supported by European FP7 ICT project CHRIS).

Stephane Lallee, Tijn van Der Zant and Peter Ford Dominey are with the Stem Cell & Brain Research Institute, INSERM U 846, Bron, France. (stephane.lallee@inserm.fr; robotijn@gmail.com; peter.dominey@inserm.fr).

Severin Lemaignan is with LAAS, CNRS, Toulouse, France. (severin.lemaignan@laas.fr) Alexander Lenz, Chris Melhuish and Sergey Skachek are with BRL, Bristol, United Kingdoms. (alex.lenz@brl.ac.uk ; Chris.Melhuish@brl.ac.uk; Sergey.Skachek@brl.ac.uk).

Lorenzo Natale is with IIT, Genoa, Italy. (lorenzo.natale@iit.it).

Felix Warneken is with Harvard University, Cambridge, USA (warneken@wjh.harvard.edu)

I. Introduction

Cooperation is a hallmark of human cognition. Early in their development, human children begin to engage in cooperative activities with other people. Critically, from early on, children are able to cooperate in novel situations, based upon social-cognitive capacities such as representing other people's intentions, visual perspective-taking, and imitation [1, 2]. The premise of our research is that similar skills are required also for human-robot cooperation. Specifically, in the CHRIS project[1], we derive the fundamental skills which enable young children to engage in cooperative activities and implement these in an integrated system capable of running on several robotic platforms to study human-robot interactions. The current research reports on this integrated system and resulting experiments with iCub [3] and the BERT2 robot platforms.

The novelty of the current research is twofold: First, we present an on-line learning method for recognition of simple human action related to object manipulation. Some research has already been done in the area of action learning and recognition by robots [4-7], however our approach is based on detection of simple perceptual primitives that can be processed independently from the perceptual system used. Second, we demonstrate that this platform-independent architecture operates successfully on two very distinct physical robot platforms, using highly distinct perceptual systems. Finally we demonstrate that because of the perceptual abstraction in the architecture, knowledge acquired about recognizable actions on one robot can be used to recognize actions (with a completely different perceptual system) on a different robot.

II. Context: Human / Robot Cooperation

A. Cooperation requirements

Studies of human infants [2, 8, 9] show that recognizing actions is a task that gradually develops over the second and third year of life. From around 14-18 months of age, infants begin to engage in novel cooperative tasks with adults, in which they have to collaborate jointly to achieve a shared

[1] www.chrisfp7.eu

goal (such as one agent holding something in place so that another agent can manipulate the object). It has been argued that from this early age, infants are already able to represent a shared plan of action (an action plan encompassing both the child's and the partner's actions taken to bring about a certain change in the world), and are able to reverse complementary roles if necessary. In other terms, infants are taking a 'bird's eye view' on the social situation, representing not only their own actions, but both their own and the partner's actions as part of a shared plan [10]. Such a shared plan allows the child to demonstrate "role reversal" where she can take on the role of either partner in a cooperative activity. We have recently implemented this type of shared planning in robotic systems which could observe actions, attribute roles, and then use the resulting shared plan to perform the cooperative task, taking the role of either one of the two participants[11, 12]. This basic representational capacity appears to be in place in human development very early on. However, over development, children become increasingly skilled in coordinating their actions with different social partners. They start to cooperate successfully with more competent adults early in the second year of life, and gradually becoming able to cooperate also with peers around 2 years of age [9]. Importantly, cooperating in fairly simple novel situations does not require extensive learning [2]. In more challenging tasks with complementary actions that require a multi-step sequence and a goal that is not transparent, direct instructions appears to be necessary[13]. Thus, we have used spoken language in human-robot cooperation in order to make the nature of the tasks explicit, so that they can be used by the robot to learn the structure of the task [14, 15] A crucial aspect of this human cooperative behavior is the ability to observe and understand new actions in real time, during the course of observation of an ongoing cooperation. Children can be exposed to novel physical devices and within a few trails of observation, learn new actions involved in manipulating these devices [1, 2].

B. *Extracting Meaning from Perception*

Robots will have to demonstrate similar learning capabilities in order to face novel situations they will encounter in the real world. Exhaustive knowledge about the world cannot be provided *a priori* by the programmer, thus the robots need an ability to learn. An important aspect of human social life is our ability to learn from others through observation and instruction [16], which is a faster and more accurate way of acquiring knowledge about complex entities than individual learning, such as trial-and-error learning. Mandler [17] suggested that the infant begins to construct meaning from the scene based on the extraction of perceptual primitives. From simple representations such as contact, support and attachment [18] the infant could construct progressively more elaborate representations of visuospatial meaning. In this context, the physical event "collision" can be derived from the perceptual primitive "contact". Kotovsky & Baillargeon [19] observed that at 6 months, infants demonstrate sensitivity to the parameters of objects involved in a collision, and the resulting effect on the collision, suggesting indeed that infants can represent contact as an event predicate with agent and patient arguments. Siskind [20] demonstrated that force dynamic primitives of contact, support and attachment can be extracted from video event sequences and used to recognize events including pick-up, put-down, and stack based on their characterization in an event logic. Related results have been achieved by Steels and Baillie [21]. The use of these intermediate representations renders the systems robust to variability in motion and view parameters. We have used a related approach to categorize movements including touch, push, give, take and take-from in the context of link these action representations to language [22]. In the current research, we extend these ideas, so that arbitrary novel actions including *cover, uncover, take, put* and *touch* can be learned in real-time with a few examples each, based on invariant sequences of primitive events specific to each action. We subsequently demonstrate that using the same architecture, such actions can be learned on a different robot platform using an entirely different perceptual system. Finally, and perhaps most interestingly, we demonstrate that knowledge of action recognition learned on one of the robots transfers directly for successful use on the other.

III. THE CHRIS ARCHITECTURE

In order to be platform-independent, a cognitive architecture should abstract away from platform-specific representations at the lowest level possible. An overview of our architecture in this context is presented in Figure1.

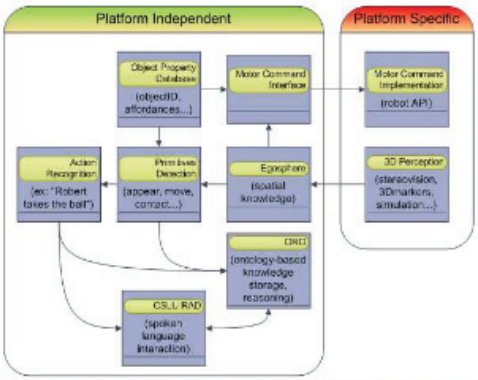


Figure 1: Overview of the Software Architecture. Each block is a stand alone software module that execute a certain function. The interface between the platform specific software and the generic architecture is the Egosphere, which allow abstraction from low level perception. Arrows represent the flow of information (data, commands), which are transported over the network via YARP. All the left part is robot independent and been tested on the iCub and BERT robots.

Robot specific components, for the 3D-perception and Motor command levels, illustrated on the right, are isolated from the rest of the system at the lowest level.

A. Scene Perception

1) EgoSphere

The first layer of abstraction between the sensory perception systems, the higher level cognitive architecture and motor control elements is formed by the EgoSphere. Unlike the sensory ego-sphere (SES) by Peters [23] which implements short term memory, associations, direction of attention in addition to localization, our simpler implementation solely acts as a fast, dynamic, asynchronous storage of object positions and orientations. The object positions are stored in spherical coordinates (radius, azimuth and elevation) and the object orientation is stored as rotations of the object reference frame about the three axes (x,y,z) of a right-handed Cartesian world-frame system. The origin of the world frame can be chosen arbitrarily and, for our experimental work, we located it at the centre of the robot's base-frame. Other stored object properties are a visibility flag and the objectID. The objectID is a unique identifier of an object which acts as a shared key across several databases (described in more detail in B below). The robot-specific 3D perception system adds objects to the EgoSphere when they are first perceived, and maintains position, orientation or visibility of these objects over time. Modules (e.g. Primitive Detection in Fig 1) requiring spatial information about objects in the scene can query the EgoSphere. No assumptions are made about the nature of an object and any further information (e.g. object name, object type) will have to be queried from the Knowledge Base using the objectID. This architecture makes the EgoSphere particularly useful for storing multi-modal information.

The EgoSphere is implemented in C++ as a client-server system using the YARP infrastructure. Software modules requiring access to the EgoSphere include a client class which provides methods like addObject(.), setObject(.), getObject(.) or getNumberOfObjects(.), etc. Clearly, at the current state, the EgoSphere is merely a convenient abstraction layer. With increasing complexity of human-robot interaction tasks during the course of our research, we plan to add further complexity (human focus of attention, confidence, timeliness etc.) whilst preserving modularity.

2) Primitive Detection

The robot should be able to recognize actions performed by other agents in order to learn, to cooperate or for safety reasons. A few systems are performing action learning and recognition [4-7, 24, 25], however none of them is completely platform independent. Since our system is taking inputs from the Egosphere, it allows applying learning and recognition algorithm that are not at all related to a specific robot. Moreover, our algorithm is using a novel approach : we have previously demonstrated [22] that actions involving change of possession could be described in term of perceptual primitives such as *contact*. Here we extend the primitives to include motion and visibility. Thus an action such as "Larry takes the ball" can be characterized in terms of a sequence of perceptual primitives:

- *Motion: Larry's hand starts to move*
- *Contact: There is a physical contact between Larry's hand and the ball*
- *Motion: Both Larry's hand and the ball start to move together and then they both stop.*

We refer to these low level events as Perceptual Primitives. Dominey & Boucher [22] demonstrated that a variety of actions could be recognized with the single primitive *contact(x,y)*. Here we extend this approach by including in addition the primitives *visible(x)* and *moving(x)*. These primitives and their corresponding arguments and truth values are computed in the Primitive Detection module, which polls the EgoSphere for changes in position and visibility. Contact is recognized by a minimum distance threshold which is determined empirically. Likewise motion is detected when the position of an object changes over an empirically determined threshold. Visibility is directly available from the EgoSphere.

3) Action Recognition

Thus, when a physical action occurs, values encoding object positions in the EgoSphere change accordingly. Primitive Detection transforms this position information into sequences of perceptual primitives. Action Recognition reads this stream of perceptual primitives and groups the elements into candidate actions. Based on empirical measures we determined that primitives which are separated by less than one sec. belong to a common action. A

primitive sequence for an action may last several seconds, but no successive primitives are separated by more than 1 sec. This limitation on fast successive actions is considered in the discussion section. When an action is performed and processed, its primitive sequence is thus segmented by the Action Recognition module, which tries to recognize it. The Action Recognition module generates and manipulates the Action Definitions database of primitive sequences as follows. It tries to match the current sequence by an exhaustive search through the database. If the sequence is not recognized, the Action Recognition module triggers the Spoken Language Interaction to ask the user for a description of the action, providing the action name, agent and object of the action. It then associates this description with the recorded sequence for future recognition. If the sequence is recognized, the Spoken Language Interface extracts the action and arguments and reports. The system thus provides object independent action recognition (i.e. if it has learned "Larry takes the ball", it is able to recognize "Robert takes the coffee-cup"). The module also detects and stores within an action definition the initial state of the objects concerned by the action, and the consequences of this action on the world (i.e. if Larry covers the ball with a box, then the ball will not be visible anymore) which will allow creation of new inference rules within the ORO module of the Knowledge Base, described below.

B. *Knowledge Base*

Through interaction with the user and the physical world, the system acquires new knowledge, and it is also initialized with certain background knowledge.

1) Object Properties Database

The OPDB is the common namespace manager for objects that can be perceived by the system. It contains physical parameters of objects, including their perceptual signature as defined by the EgoSphere. Each object that is known to the system (that can be perceived and represented in the EgoSphere) has a unique identifier (the objectID) which serves as an index into the OPDB and the Knowledge Base in general.

2) The Open Robot Ontology

ORO (the "OpenRobot Ontology" server) is the semantic layer of the system. It has been designed to integrate easily in different robotic architectures by ensuring a limited set of architecture requirements. ORO is built around a socket-based server that stores, manages, processes and exposes knowledge. ORO is portable (written in Java), and can be easily extended with plug-ins, making it suitable to new applications. In the frame of the CHRIS project, a YARP bridge has been added, thus exposing the ORO RPC methods in a network-transparent way. ORO relies internally on the OWL ontologies dialect to store knowledge as RDF triples. It uses the open-source Jena[2] RDF graph library for storage and manipulation of statements and the equally open-source Pellet[3] first order logic reasoner to classify/apply rules and compute inferences on the knowledge base.

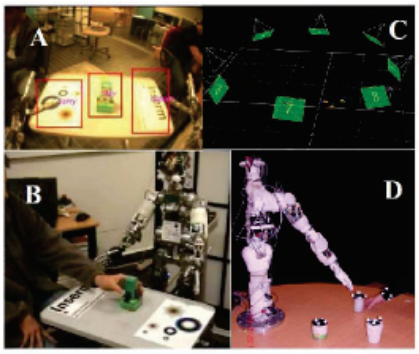


Figure 2: Specific Robotic Platforms. A. Vision processing using Spikenet™ with the video image from the iCubLyon01 robot, pictured in B. C. The Vicon™ configuration for visual perception with the Bert Robot, pictured in D.

Besides simply storing and reasoning about knowledge, ORO offers several useful features for human-robot interaction: events registration (e.g. "Tell me when any kind of tableware appears on the table."), categorization capabilities, independent cognitive models for each agent the robot knows and different profiles of memory (short-term, episodic, long-term). The server loads an initial ontology at startup, the so-called OpenRobots Common-Sense Ontology. This initial ontology contains a set of concepts (over 400 in the last version), relationships between concepts and rules that defines the cultural background of the robot, i.e. the concepts the robot knows a priori. This common-sense knowledge is very focused on the requirement of our scenarios, namely, human-robot interaction with some well-known everyday objects (cups, cans, etc.). It contains as well broader concepts like agents, objects, location, etc. The common-sense ontology relies heavily on the de-facto standard OpenCyc upper-ontology for the naming of concepts, thus ensuring a good compatibility with other knowledge sources (including Internet-based ones, like WordNet[4] or DBPedia[5]). The ontology then dynamically evolves as the robot acquires new facts: these are provided either from the EgoSphere via the primitive detection module, or via spoken language interaction with human.

3) Action Definitions

Actions that have been learned are stored in the Action Definitions Database. Actions are defined in terms of three

[2] http://jena.sourceforge.net

[3] http//clarkparsia.com/pellet
[4] http://wordnet.princeton.edu
[5] http://dbpedia.or

types of information. The Enabling State defines the state of the objects involved in the action before the action takes place. The Primitive Sequence is the time ordered set of primitive events that make up the dynamic component of the action. Finally, the Resulting State is the (potentially) new state of affairs after the action is completed. The action recognition capability described above relies primarily on the Primitive Sequence for action recognition.

C. Spoken Language Interaction

The spoken language interaction is provided by the CSLU Toolkit [26] 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. Our system is thus state based, with the user indicating the nature of the current task (including whether he wants interact in the context of object recognition, action recognition or action sequence recognition tasks). In each of these subdomains, the user can then indicate that he is ready to show the robot a new example (object, action or action sequence) and the robot will attempt to recognize or learn what is shown. RAD scripts are in done in TCL which allows communication of speech data to other modules through YARP.

D. YARP

Software modules in the architecture are interconnected using YARP [27], 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.

Finally, YARP is written in C++, so it is normally used as a library in C++ code. However, any application that has a TCP/IP interface can talk to YARP modules using a standard data format. Within the CHRIS project this turned out to be of fundamental importance as it allowed to "glue" together different applications (e.g. the RAD toolkit, the ORO server or the VICON system) into a single integrated, working system.

IV. Integration Platforms

The CHRIS Software Architecture has been successfully tested on two different platforms illustrated in Figure 2.

A. Platform iCubLyon01

1) Robot Platform

The iCub [3] is an open-source robotic platform shaped as three and a half year-old child (about 104cm tall), with 53 degrees-of-freedom (DOF) distributed on the head, arms, hands and legs. The current work was performed on the iCubLyon01 at the INSERM laboratory in Lyon, France. The DOF are distributed over the full body: 6 for the head, 3 for the waist, 6 in each leg and 7 for each arm. The iCub has been specifically designed to study manipulation, for this reason the number of DOF 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. All the code and documentation is provided open source by the RobotCub Consortium, together with the hardware documentation and CAD drawings. The robot hardware is based on high-performance electric motors controlled by a DSP-based custom electronics. From the sensory point of view the robot is equipped with cameras, microphones, gyroscope, position sensors in all joints, force/torque sensors in each limb.

2) 3D Spatial-Temporal Object Perception

The iCubLyon01 platform employs vision based perception operating on the image streams from the robot's stereo cameras. Objects are recognized based on detection of predefined object templates using the commercial system Spikenet [28]. It uses a spiking neural network technology to provide fast recognition of objects in an image. By doing this with the two stereo cameras of the robot, we can estimate the Cartesian coordinates of the objects and feed the EgoSphere. To do so, a simple wrapper around the Spikenet API is used for retrieving the camera images, processing them and broadcasting the results over the network via YARP. Another module is then used to read this data, filter the noise and update the EgoSphere appropriately. Once in the EgoSphere, the spatial-temporal object information is platform-independent.

B. Platform BERT2 BRL

1) Robot Platform

BERT2 (Bristol-Elumotion-Robot-Torso-2) is an upper-body humanoid robot designed, and currently still under construction, at Bristol Robotics Laboratory in close co-operation with their mechanical engineering partner Elumotion[6]. The torso comprises four joints (hip rotation, hip flexion, neck rotation and neck flexion). Each arm is equipped with 7 DOF. The wrist provides a mounting interface for a sophisticated humanoid hand or a simple gripper. Each of these 18 joints is actuated by a brushless DC motor via a Harmonic Drive (TM) gear box. One of the

[6] www.elumotion.com

main motivations that guided the design of BERT2 was the suitability to interact with humans safely and naturally using Expressive Face and Gaze Tracking. One important non-verbal communication channel we have focused on is facial expression with a particular emphasis on gaze, as used in human-human interaction [29].

2) 3D Spatial-Temporal Object Perception

The BERT2 platform uses the VICON motion capture system (with 8 stationary IR cameras) and light reflective markers arranged into unique patterns, to distinguish between scene objects and to detect their position and orientation in 3D space. This provides reliable and robust 360 degree scene perception. The human interacting with the robot also wears a garment equipped with markers, thus body positions and postures are also available to the robot. There are several layers of abstraction in BERT2 VICON perception. At the lowest level there is VICON hardware and software together with VICON object and actor model templates, which store information about the marker topology of the objects to be captured. The VICON software broadcasts this captured data on the network, using TCP/IP. This data is picked up by the module "ViconLink", which is an easily reconfigurable data bridge between the VICON software and the YARP framework. The next layer of abstraction is the "Object Provider" module. Its main purpose is to update the EgoSphere with the most recent object positions and to filter the noise in the VICON data. Again, once in the EgoSphere, the spatial-temporal object information is platform-independent.

V. Experiments

Diverse experiments have been performed in a distributed manner on the two platforms. The first goal of these experiments is to show the portability of the full cognitive system between multiple robots, more than giving precise benchmarks of the skills provided by the system. The experiments reported on here are those which were run on the iCub and BERT2 platforms using the identical CHRIS architecture (see accompanying video).

A. Object learning

The goal of the experiment is to allow the user to teach the system the name and properties of new objects. In these experiments, two sets of objects have been pre-specified respectively for each of the two 3D perception systems. This corresponds to visual templates for Spikenet on the iCub, and reflective marker topologies for VICON on BERT2. Initially the objects can thus be recognized and tracked, but they have no associated semantics. In the experiment, the human moves an object to indicate the focus to the robot, which then asks for the name and the type of the object. Learning the object's type (i.e. "cup") links its semantics to the other concepts the robot already knows, including initial commonsense knowledge from ORO. When an object moves, the platform specific perception systems identify and accurately localize the object. The respective object perception module then updates the EgoSphere in real time. At this point we are entering the platform-independent CHRIS architecture. The Primitive Detection module regularly polls the EgoSphere for visibility and object coordinates, and sends extracted primitives to other interested modules. In this case, it sends to ORO a notification when an object starts or stops moving. In parallel, the Spoken Language Interaction system manages the verbal human-robot interaction. It queries ORO to know which objects are currently moving and if the names of these objects are known. If they are unknown then it asks to the human for more information as described in the dialog below.

- [Robot] Initializing... *about 5 sec* ...What's next?
- [Human] *moves an object*
- [Robot] *does not know the object* What is it called?
- [Human] coffee cup
- [Robot] Did you say coffee cup?
- [Human] yes
- [Robot] Ok. Now I know. What kind of object is coffee cup?
- [Human] a cup
- [Robot] Did you say cup?
- [Human] yes
- [Robot] So coffee cup is a cup. What's next?

During this interaction, two new statements are added to the ORO database. The first associates the objectID for the recognized object with the label "coffee cup". The second associates this objectID with the type "cup" which allows the system to exploit inheritance of knowledge about cups from the preloaded ontology. The robot is then able to use this additional knowledge in order to reason and describe objects. The next time a human shows the coffee cup, the robot will say "this is a coffee cup". In addition, we could ask to the robot "could you show me a cup?" to have the robot to point to the coffee cup or to a tea cup, exploiting the type information. This naming experiment was successfully completed for multiple objects on the iCub and BERT2 platforms.

B. Action learning

As described above, an important skill in cooperation is the robot's ability to understand its partner's actions, or the actions of two humans cooperating while the robot learns by observing. In the "watch action" mode, the robot tries to recognize and describe the actions it sees. It can then store the recognized actions in a plan that can be executed later, or it can ask for more information if the observed action is not known or uncertain. Regarding the flow of information through the system, it is the same as in the object naming experiment but it now includes the Action Recognition module. Under control of the Spoken Language Interface, Action Recognition receives primitive events from Primitive

Detection. It compares the segmented primitive event sequences with known actions in the Action Database, and provides in real time actions that are recognized or a special event if an action has been segmented but not recognized. These recognitions are then processed by Spoken Language Interaction which manages the situation either by announcing the name and arguments of the recognized action, or prompting the user to describe the action and its arguments, as illustrated in this dialog excerpt.

- [Robot] Initializing... *about 5 sec* ...What's next?
- [Human] Watch my actions
- [Robot] Ok. Show me something
- [Human] *puts a coffee cup on the table*
- [Robot] What happened?
- [Human] John puts the coffee cup
- [Robot] Ok. Show me something
- [Human] *removes the coffee cup from the table*
- [Robot] What happened?
- [Human] John takes the coffee cup
- [Robot] Ok. Show me something
- [Human] *put a book on the table*
- [Robot] I saw John puts the book

At this point the robot is able to recognize the two simple actions *put* and *take* independently of the object targeted by the action. In this scenario a single demonstration was sufficient. In practice the robot may need to see the same action several times before being able to recognize it. Lallee et al [30] preformed extensive testing of this system on the iCubLyon01 platform. In over 100 action presentations, with the actions *cover, uncover, put, take* and *touch,* on average the system required less than three examples to correctly learn a given action so that it could subsequently be recognized without error. The crucial experiment here involved performing the same action learning tests on the BERT platform, where visual perception based on pattern matching with Spikenet would be replaced by reflective marker tracking provided by VICON. We tested BERT with the actions *put*, *take*, and *touch*. These actions were successfully learned, and generalized to new objects. This indicates that by abstracting 3D spatial-temporal information in the EgoSphere, the CHRIS architecture is indeed platform-independent. Our final experiment replies to the question "can knowledge about the spatial-temporal characteristics of an action learned on one platform be used for action recognition on another?"

C. *Knowledge transmission between Robots*

Following an interaction session with humans, the robot Knowledge Base acquires new knowledge (of object and action definitions) through learning. This acquired knowledge is stored prior to system shutdown and reloaded at subsequent system startup, thus allowing progressive accumulation of experience over extended time. In the current experiment, we took the Action Recognition database that was generated while actions were being learned on BERT, and loaded it at startup on the iCubLyon01. We then tested the Action Recognition capability, by performing *put* and *take* actions. In a set of 20 trials (10 each for *put* and *take*) we observed an overall recognition accuracy of 85%. The errors were due to noise in the vision system which produces false indications of motion (see discussion). Importantly, the iCub was able to recognize actions that had been learned on BERT, thus exploiting the experience of a different robot.

VI. Discussion

We present an architecture that exploits the idea of abstracting the cognitive architecture from the robot specific body and sensors. It should be noted that the cognitive function of the robot can still be considered embodied as the architecture acquires all its information from interaction between the robot and the world, via the low level abstraction of the EgoSphere. Thanks to this abstraction, we were able to provide to different robots the same high level capabilities for perception and reasoning, and to share knowledge acquired via different sensors.

A. *Limitations and future development:*

The work described here emphasizes abstraction at the sensory level (and does not address motor control), by requiring a common format for spatial input to the system from diverse sensors. A parallel approach is to be taken at the motor command level (Motor Command Interface, Fig 1). This is based on the definition of a set of actions including *give, take, put, point* and their arguments. Their initial and final states are defined in a platform independent manner, but the specific joint-level implementation is specified in the context of the corresponding robot platforms. This will provide a capability consistent with that described by Demiris & Johnson [31] where action execution and performance can mutually benefit from shared representations. Action Recognition provides real-time formation and recognition of sequential patterns of primitive events (motion, visibility and contact) specific to different actions. It is thus sensitive to noise in the 3D perception sensors. We are currently rendering this approach more robust. This includes the use of a probabilistic approach for matching the segmented primitive event sequences with the learned actions, optimization of spatio-temporal filtering to reduce false motion from visual jitter, and inclusion of the initial-to-final state transitions as additional components in definition of an action. Likewise, in the current version, successive actions (e.g. taking an object, then putting it at a new location) should be separated by at least one second, so that the system can automatically distinguish and segment the perceptual primitive sequences. This is consistent with our current constraint that when demonstrating action, users show actions one after another, and wait to see if the robot recognizes, before proceeding. Future work will address

more fluent action sequences in the context of learning from demonstration [32].The speech that we have used here is relatively primitive and sometimes ungrammatical. We have previously explored the more extensive possibilities of relating the argument structure of grammatical sentences to the argument structure of actions in terms of execution [15, 33, 34]. We are now extending these approaches to action observation and description with the use of more appropriate grammar.

B. Conclusions

While robotic platforms are becoming increasingly complex, the development of cognitive systems can be advanced by the development of more standard ways to access the sensory-motor layer. Our system independent architecture contributes to the deployment of cognitive abilities on diverse robot platforms that can interface with the abstraction layer defined by the EgoSphere and the motor command interface. We believe that the continued development of increasingly well defined and standard interfaces between robot platforms and cognitive system can accelerate the development of robot intelligence, and we are taking a first step in that direction. In doing so we have also taking the first steps towards the idea of having different learning machines (the robots individuals) updating and sharing a common global knowledge base, thus leveraging experience from multiple sources [21]. Further work will investigate methods to enhance this ability and to allow robot platforms distributed over the world to take advantage of it.

VII. Acknowledgment

This research was supported by the European Commission under the Robotics and Cognitive Systems, ICT Project CHRIS (FP7-215805).

VIII. References

[1] Tomasello, M., et al., *Understanding and sharing intentions: The origins of cultural cognition.* Behavioral and Brain Sciences, 2005. **28**(05): p. 675-691.

[2] Warneken, F., F. Chen, and M. Tomasello, *Cooperative activities in young children and chimpanzees.* Child Development, 2006. **77**(3): p. 640-663.

[3] Metta, G., et al. *The iCub humanoid robot: an open platform for research in embodied cognition.* in *PerMIS: Performance Metrics for Intelligent Systems Workshop*. 2008. Washington DC, USA.

[4] Yamato, J., J. Ohya, and K. Ishii. *Recognizing human action in time-sequential images using hidden Markov model.* in *IEEE Proc. Computer Vision and Pattern Recognition*. 1992.

[5] Johnson, M. and Y. Demiris, *Perceptual perspective taking and action recognition.* International Journal of Advanced Robotic Systems, 2005. **2**(4): p. 301-308.

[6] Bobick, A. and Y. Ivanov. *Action recognition using probabilistic parsing*. in *IEEE Proc. Computer Vision and Pattern Recognition*. 1998.

[7] Demiris, Y. and B. Khadhouri, *Hierarchical attentive multiple models for execution and recognition of actions.* Robotics and Autonomous Systems, 2006. **54**(5): p. 361-369.

[8] Warneken, F. and M. Tomasello, *Helping and cooperation at 14 months of age.* Infancy, 2007. **11**(3): p. 271-294.

[9] Brownell, C., G. Ramani, and S. Zerwas, *Becoming a social partner with peers: Cooperation and social understanding in one-and two-year-olds.* Child Development, 2006. **77**(4): p. 803-821.

[10] Carpenter, M., M. Tomasello, and T. Striano, *Role reversal imitation and language in typically developing infants and children with autism.* Infancy, 2005. **8**(3): p. 253-278.

[11] Dominey, P. and F. Warneken, *The basis of shared intentions in human and robot cognition.* New Ideas in Psychology, 2009: p. (in press).

[12] Lallée, S., F. Warneken, and P. Dominey. *Learning to collaborate by observation*. in *Epirob*. 2009. Venice.

[13] Ashley, J. and M. Tomasello, *Cooperative problem-solving and teaching in preschoolers.* Social Development, 1998. **7**(2): p. 143-163.

[14] Dominey, P., et al. *Anticipation and initiative in human-humanoid interaction.* in *International Conference on Humanoid Robotics*. 2008.

[15] Dominey, P., A. Mallet, and E. Yoshida. *Real-time cooperative behavior acquisition by a humanoid apprentice.* in *International Conference on Humanoid Robotics*. 2007. Pittsburg, Pennsylvania.

[16] Tomasello, M. and A. Whiten, *The cultural origins of human cognition*. 1999: Harvard University Press Cambridge, MA.

[17] Mandler, J., ed. *Preverbal representation and language*. Language and space. 1996, MIT Press. 365--384.

[18] Talmy, L., *Force dynamics in language and cognition.* Cognitive science, 1988. **12**(1): p. 49-100.

[19] Kotovsky, L. and R. Baillargeon, *The development of calibration-based reasoning about collision events in young infants.* Cognition, 1998. **67**(3): p. 311-351.

[20] Siskind, J., *Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic*. Journal of Artificial Intelligence Research, 2001. **15**(1): p. 31-90.

[21] Steels, L. and J. Baillie, *Shared grounding of event descriptions by autonomous robots.* Robotics and Autonomous Systems, 2003. **43**(2-3): p. 163-173.

[22] Dominey, P. and J. Boucher, *Learning to talk about events from narrated video in a construction grammar framework.* Artificial Intelligence, 2005. **167**(1-2): p. 31-61.

[23] Peters, I.R.A., K.A. Hambuchen, and R.E. Bodenheimer, *The sensory ego-sphere: a mediating interface between sensors and cognition.* Auton. Robots, 2009. **26**(1): p. 1-19.

[24] Kaiser, M. and R. Dillmann. *Building elementary robot skills from human demonstration.* in *Proceedings of the International Conference on Robotics and Automation*. 1996.

[25] Nicolescu, M. 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*. 2003. Melbourne: ACM.

[26] Sutton, S., et al. *Universal speech tools: The CSLU toolkit*. in *Fifth International Conference on Spoken Language Processing*. 1998.

[27] Fitzpatrick, P., G. Metta, and L. Natale, *Towards Long-Lived Robot Genes.* Robotics and Autonomous Systems, 2007. **56**(1): p. 29-45.

[28] Thorpe, S., et al., *SpikeNet: Real-time visual processing with one spike per neuron.* Neurocomputing, 2004. **58**: p. 857-864.

[29] Senju, A. and G. Csibra, *Gaze following in human infants depends on communicative signals.* Current Biology, 2008. **18**(9): p. 668-671.

[30] Lallée, S., et al., *Linking language with embodied teleological representations of action for humanoid cognition.* Frontiers in Neurobotics (submited), 2010.

[31] Demiris, Y. and M. Johnson, *Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning.* Connection Science, 2003. **15**(4): p. 231-243.

[32] Argall, B., et al., *A survey of robot learning from demonstration.* Robotics and Autonomous Systems, 2009. **57**(5): p. 469-483.

[33] Dominey, P., A. Mallet, and E. Yoshida. *Progress in programming the hrp-2 humanoid using spoken language*. in *IEEE International Conference on Robotics and Automation*. 2007.

[34] Dominey, P., A. Mallet, and E. Yoshida, *Real-Time spoken-language programming for cooperative interaction with a humanoid apprentice.* Intl J. Humanoids Robotics, 2009. **6**(2): p. 147-171.

Appendix 2

Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions

Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions

Stephane Lallée, Ugo Pattacini, Jean David Boucher, Séverin Lemaignan, Alexander Lenz, Chris Melhuish, Lorenzo Natale, Sergey Skachek, Katharina Hamann, Jasmin Steinwender, Emrah Akin Sisbot, Giorgio Metta, Rachid Alami, Matthieu Warnier, Julien Guitton, Felix Warneken, Peter Ford Dominey

Abstract— If robots are to cooperate with humans in an increasingly human-like manner, then significant progress must be made in their abilities to observe and learn to perform novel goal directed actions in a flexible and adaptive manner. The current research addresses this challenge. In CHRIS.I [1], we developed a platform-independent perceptual system that learns from observation to recognize human actions in a way which abstracted from the specifics of the robotic platform, learning actions including "put X on Y" and "take X". In the current research, we extend this system from action perception to execution, consistent with current developmental research in human understanding of goal directed action and teleological reasoning. We demonstrate the platform independence with experiments on three different robots. In Experiments 1 and 2 we complete our previous study of perception of actions "put" and "take" demonstrating how the system learns to execute these same actions, along with new related actions "cover" and "uncover" based on the composition of action primitives "grasp X" and "release X at Y". Significantly, these compositional action execution specifications learned on one iCub robot are then executed on another, based on the abstraction layer of motor primitives. Experiment 3 further validates the platform-independence of the system, as a new action that is learned on the iCub in Lyon is then executed on the Jido robot in Toulouse. In Experiment 4 we extended the definition of action perception to include the notion of agency, again inspired by developmental studies of agency attribution, exploiting the Kinect motion capture system for tracking human motion. Finally in Experiment 5 we demonstrate how the combined representation of action in terms of perception and execution provides the basis for imitation. This provides the basis for an open ended cooperation capability where new actions can be learned and integrated into shared plans for cooperation. Part of the novelty of this research is the robots' use of spoken language understanding and visual perception to generate action representations in a platform independent manner based on physical state changes. This provides a flexible capability for goal-directed action imitation.

Manuscript received March 15, 2011. This work was fully supported by European FP7 ICT project CHRIS). Stephane Lallee, Jean-David boucher and Peter Ford Dominey are with the Stem Cell & Brain Research Institute, INSERM U846, Bron, France. (stephane.lallee@inserm.fr). Severin Lemaignan, Emrah Akin Sisbot and Rachid Alami are with LAAS, CNRS, Toulouse, France. Alexander Lenz, Chris Melhuish and Sergey Skachek are with BRL, Bristol, United Kingdom. Ugo Pattacini,Lorenzo Natale and Giorgio Metta are with IIT, Genoa, Italy. Jasmin Steinwender and Katharina Hamann are with Max Planck Institute, Leipzig, Germany. Felix Warneken is with Harvard University, Cambridge, USA

I. Introduction

For embodied agents that perceive and act in the world, there is a strong coupling or symmetry between perception and execution which is constructed around the notion of goal directed action. Hommel et al [2] propose a philosophy for the cognitive mechanisms underlying perception and action – the Theory of Event Coding. According to this theory, the stimulus representations underlying action perception, and the sensorimotor representations underlying action are not coded separately, but instead are encoded in a common representational format. In this context 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 [3]. This research corroborates the emphasis from behavioral studies on the importance of the goal (rather than the details of the means) in action perception [4].

Within a sensorimotor architecture a number of benefits derive from such a format, including the direct relation between action perception and execution that can provide the basis for imitation. This is consistent with our previous research in the domain of robot perception and action in the context of cooperation ([5, 6]). The current research extends our previous work on the learning of composite actions by exploiting this proposed relation between action execution and perception. Part of the novelty of the current research is that the action repertoire is open: the robot can learn new actions in both dimensions of perception and execution. The learned actions take arguments including agent, object and recipient. Maintaining this symmetry of action perception and execution lays the framework for imitation and the use of imitation in cooperation [5, 6].

We look to human development to extract requirements on how to implement such an action representation. In this context, two important skills for infants are the ability to detect an action as being goal directed and to determine its

agency. Studies of infant action perception [4, 7] have led to the extraction of a core set of conditions which allows the infant to identify goal-directed actions. In the current research, we implement in our system the ability to address aspects of these human requirements both in terms of perception (detect and represent *salient actions effects*) and execution (ability to achieve a goal through an action using *equifinal variations*). We demonstrate how those capabilities can be used by the robot to imitate or mirror human actions (which involve both recognition and execution) in a way that should match the human requirements for goal attribution.

Learning by imitation is a major area of research in robot cognition today [8-12]. Our novel contribution to this domain is the encoding of action in terms of perceptual state changes and composed motor primitives that can achieve these state changes, in a manner that allows the robot to learn new actions as perception – execution pairs, and then use this knowledge to perceive and imitate. These actions can take several arguments, e.g. `AGENT put the OBJECT on the RECIPIENT`. This allows for the generalization of learned actions to entirely new contexts, with new objects and agents. In our long-term research program, this provides the basis for learning to perform joint cooperative tasks purely through observation.

II. Context: Goal directed actions

A. Goals attribution requirements

Studies of human infants [4, 13-15] indicate that their ability to determine the goal of an action begins to develop between 6 and 9 months, demonstrated by the ability of infants to encode behaviors such as a hand grasping for an object as being directed at the goal-object rather than encoding the hand's specific movement. An important issue that has been discussed within the field is the difference between actions that are familiar to the infant and more unfamiliar actions which may not include human features (like a robotic gripper grasping a toy). Woodward [14] initially argued that only observed actions that the infant is able to execute herself are represented as goal-directed. However later studies [4, 7] demonstrated that indeed infants are able to attribute goal directedness for novel actions early assuming two conditions: first the action has to produce a salient effect on the world state (like the motion from one place to another). The second condition is that the agent is able to achieve the same state change in different ways (such as avoiding an obstacle instead of using a straight trajectory), in other words the action is demonstrated to possess equifinal variations.

B. Implementing those requirements

Our implementation of action, both in the context of perception from CHRIS.I [1] and execution is based on actions as state changes. One of the strong implications of this is the equifinality of action. That is, the same action "put the box on the toy" may be realized in a variety of ways (with one hand, or the other) but with the equivalent final outcome, one of the key characteristics that allow action to be considered goal directed. If the robot is able to demonstrate equifinal means of achieving his actions, then humans may be more likely to attribute a goal to them. This assumption has been shown to be true in infants [4, 16] and would need to be tested on adults, however assuming the fact that all our teleological system seems to be built on those core capabilities it is likely that a benefactor effect could be found also on adults.

In our action recognition system [1] we exploited Mandler's [17] suggestion that the infant begins to construct meaning from the scene based on the extraction of perceptual primitives. From simple representations such as contact, support and attachment [18] the infant could construct progressively more elaborate representations of visuospatial meaning. In this context, the physical event "collision" can be derived from the perceptual primitive "contact". Kotovsky & Baillargeon [19] observed that at 6 months, infants demonstrate sensitivity to the parameters of objects involved in a collision, and the resulting effect on the collision, suggesting indeed that infants can represent contact as an event predicate with agent and patient arguments.

In this paper we describe an evolution of the action recognition system described in [1]. This new system is still based on sequences of perceptual event primitives (visibility, motion, contact), however those primitives are now represented in terms of the impact they have on the world state. Primitives can be queued and their effects added so that a sequence of them will be a way to reach an end state from an initial state. If a sequence produces no change in the world state, then it will not be taken into account by the system, which mimics the ability of children to emphasis actions that produce a salient effect on the world. This rejection of "useless" actions allow the system to be more stable: for example an object which appears and then disappears quickly may be only a false recognition of the perceptual system.

These requirements are implemented on both the perceptual and executive components of the system. In CHRIS.I [1] we presented a system architecture for cooperation. Here we zoom in on the action related components which handle the complete link from perception to motor commands in term of actions.

III. Experimental Platforms

A crucial aspect of our research is that the architecture should allow knowledge acquired on one robot to be used on physically distinct platforms. In the current study this is demonstrated using two different version of the iCub platform in Lyon France, and Genoa Italy, respectively, and the Jido robot in Toulouse, France.

The iCub [20] is an open-source robotic platform shaped as three and a half year-old child (about 104cm tall), with 53 degrees of freedom distributed on the head, arms, hands and legs. 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. All the code and documentation is provided open source by the RobotCub Consortium, together with the hardware documentation and CAD drawings. The robot hardware is based on high-performance electric motors controlled by a DSP-based custom electronics. From the sensory point of view the robot is equipped with cameras, microphones, gyroscopes, position sensors in all joints, force/torque sensors in each limb.

While both iCubs are instances of the iCub, they are distinct in the implementation of motor control as the iCubGenoa01 is equipped with force sensors that allow force control; the iCubLyon01 is only controlled in velocity and position modes. Thus, the essential role of the motor primitive pool as the common abstraction layer across robots is maintained. Jido, on the other hand is an entirely different robot, which allows us to truly explore the platform independence of our system.

Jido is a fully-equipped mobile manipulator that has been constructed in the framework of Cogniron (IST FET project: www.cogniron.org). Jido, a MP-L655 platform from Neobotix, is a mobile robot designed to interact with human beings. It is presented on figure 3. Jido is equipped with: (i) a 6-DOF arm, (ii) a pantilt unit system at the top of a mast (dedicated to human-robot interaction mechanisms), (iii) a 3D swissranger camera and (iv) a stereo camera, both embedded on the pan tilt unit, (v) a second video system fixed on the arm wrist for object grasping, (vi) two laser scanners, (vii) one panel PC with tactile screen for interaction purpose, and (viii) one screen to provide feedback to the robot user. Jido has been endowed with functions enabling to act as robot companion and especially to exchange objects with human beings. So, it embeds robust and eþ cient basic navigation and object recognition abilities.

IV. The CHRIS Architecture – Focus on Action

In order to be platform-independent, action representation is abstracted from platform-specificities at the lowest level possible. An overview of the CHRIS architecture in this context is presented in Figure1.

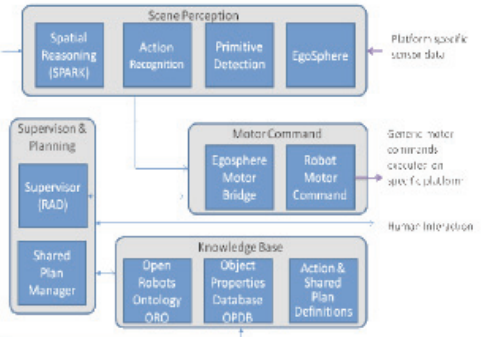


Figure 1: CHRIS Architecture. Arrows represent the flow of information (data, commands), which are transported over the network via YARP. Perceptual information enters Scene Perception. Object positions from Egosphere are processed by Primitive Recognizer and Action Recognizer for learning and recognition, and enter SPARK for inference of spatial relations which are stored in ORO. Shared Plan Manager links perceputal and exectutive action representations and plans. Supervisor manages HRI, the learning of new action execution, and verificiation from ORO that execution preconditions hold.

A. *Abstraction of Action Perception and Execution*

Two layers of abstractions are required in order to have a platform independent architecture: perceptual and motor. Both of them rely on the Egosphere module.

1) Scene Perception

The first layer of abstraction between the sensory perception systems and the higher level cognitive architecture and motor control elements is formed at the level of the Egosphere which serves as a fast, dynamic, asynchronous storage of object positions and orientations. The object positions are stored in spherical coordinates (radius, azimuth and elevation) and the object orientation is stored as rotations of the object reference frame about the three axes (x,y,z) of a right-handed Cartesian world-frame system. The origin of the world frame can be chosen arbitrarily and, for our experimental work, we located it at the centre of the robot's base-frame. Other stored object properties are a visibility flag and the objectID. The objectID is a unique identifier of an object which acts as a shared key across several databases (see [1] for details). The robot-specific 3D perception system adds objects to the Egosphere when they are first perceived, and maintains position, orientation or visibility of these objects over time. Modules requiring spatial information about objects in the scene can query the Egosphere. The Egosphere is

implemented in C++ as a client-server system using the YARP infrastructure. Software modules requiring access to the Egosphere include a client class which provides methods like addObject(), setObject(), getObject() or getNumberOfObjects(), etc. The Egosphere is thus a convenient abstraction layer. With increasing complexity of human-robot interaction tasks during the course of our research, we will add further complexity (human focus of attention, confidence, timeliness etc.) whilst preserving modularity. This is exemplified by the spatial reasoning (e.g. visibility by line of sight) provided by Spark. Within the Jido platform-independent component, the functionality of the EgoSphere is preserved within Spark.

2) Perceptual Primitives, Events and State Changes

The action recognition capability is based on the extraction of meaningful primitive events from the flow of object positions and visibilities represented in the Egosphere and Spark. Again we based our system findings from developmental psychology. We implemented perceptual primitives similar to those described in [21-25]. We have previously used this primitives based approach in [26, 27] and we identified a core set of primitive events that are simple and provide a solid basis for action construction. There are six primitive event divided in three categories:

- Visibility (object appears or disappears)
- Motion (object starts or stops moving)
- Contact (contact made or broken between 2 objects)

Each of these primitive event is coded in terms of the state change it effects on the world (e.g: if an object appears, *visibility(object)* will be added to the world state). The Primitive Recognizer extracts those 6 primitives by constantly monitoring the Egosphere. It then broadcasts the detected events to the Action Recognizer.

3) Motor Primitives

The current research extends this notion of compositionality for action perception from CHRIS.I [1] to action execution. As for the perceptual system, the action execution system requires a suitable abstraction that provides a platform independent interface to the robot motor capabilities. Motor primitives rely on the idea that complex motor tasks may be achieved by the combination of simple parameterized controllers we call primitives. This framework is consistent with studies of biological motion [28], which demonstrate that motion of biological beings is achieved by high level motor commands triggering a sequence of motor primitives leading finally to an effective motion of the muscles. Using hierarchies of primitives for control in robotics is becoming a widely used method [29-36]. In our approach, what we call a Motor Primitive is already a symbolic action. The implementation of those actions is robot specific, what is important is that all robots share the same motor interface, as a pool of Motor Primitives. In the current system the primitives that are implemented on the robot are:

- Grasp (object)
- Release (location)
- Touch (object)
- Look-At (object)

We do not claim the completeness of this pool for all possible interactions, but these primitives were sufficient in the context of robot and human interaction through manipulation of objects on a table. The arguments for these primitives are objects whose Cartesian coordinates are recovered from the Egosphere.

B. Action Representation

The concept of Action and its representation is at the center of our architecture. Inspired by the perception-execution symmetry [2] we impose the requirement that the same data structure shall accommodate both the perceptual and executive components of action. It also includes teleological information, that is, the state changes that are induced by that action.

1) Action Representation for Perception

Our representation of action started with a purely perceptual definition [1, 6, 37]. Specifically the Action Recognizer module is constantly monitoring the flow of perceptual primitives sent by the Primitive Recognizer module. We make the assumption that two actions will be separated by a temporal delay, so we can use this delay to segment meaningful sequences of primitives. When such an independent sequence is detected, it is tagged as being a potential action which is then evaluated by the recognition process. The action data structure is similar to that for events since actions are composed of primitive events, and both produce a salient change (or changes) in the world state. The Action Recognizer stores a list of all the known actions and compares them with the incoming potential actions. All the primitives contained in the received sequence are added so that the global world state change of this sequence is obtained, then if a known action creates the same change in the environment it is recognized as being the observed action. We have to stress the fact that this "world change" is argument independent: if the system has learnt an action *cover(object A, object B)* then it will recognize a *cover(toy,box)* as well as a *cover(bowl, plate)*.

Actions possess characteristics in addition to those of event primitives. The state change produced by an event primitive is called post-condition, because it is applied after the primitive occurred. In addition to post-conditions an action has pre-conditions which can either allow or prevent it to occur (for example covering the bowl needs the bowl to be visible and uncover the bowl needs the bowl to be covered). Those pre/post conditions are a useful mechanism that allows forward/backward chaining and finally teleological reasoning (see [37] for more details about this aspect). Actions also contain a field describing the executing

agent. Agency detection is based on motion primitives associated with human hands that are detected using the Kinect device which provides information about human hands to the Egosphere (see below).

2) Action Representation for Execution

In order to bridge the gap between perception and execution, the Shared Plan Manager module combines motor representations with perceptual representations of action. While we currently address the learning of single actions as the simplest motor plans, the system is designed to naturally extend to more complex shared plans, based on our earlier work [6].

When the user asks the robot to perform an action the Shared Plan Manager searches for a plan with that name. If no such plan is found, then the Shared Plan Manager asks the user to enumerate the motor primitives (described above) that constitute that action.

The system can thus learn to perform complex actions such as put `the box on the toy` as a composite sequence of `grasp box, release box on toy`G We implement a form of argument binding so that this newly learned action can generalize across all objects. That is the robot can then perform the action `put the toy on the table`.

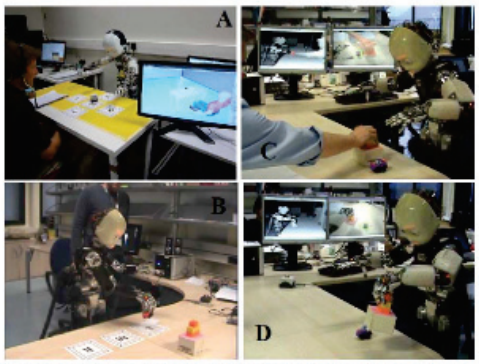


Figure 2: Experiments on iCubLyon01 and iCubGenoa01. A. Experiments 1 and 2 where human teaches robot new actions. Note in right foreground the representation of the spatial environment in SPARK. B. Replication of actions learned in Lyon with iCubLyon01 transferred to iCubGenoa01 in Genoa. C. Human demonstrates the "cover the toy with the box" action, and the iCubGenoa01 recognizes and imitates that action.

C. Supervision

Action perception and execution are coordinated by the HRI Supervisor. The Supervisor manages spoken language interaction with the CSLU Toolkit [38] 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. Our system is thus state based, with the user indicating the nature of the current task (including whether he wants interact in the context of action recognition, execution or imitation tasks). In each of these subdomains, the user can then indicate that he is ready to show the robot a new example and the robot will attempt to recognize, perform or learn what is shown.

A principal function of the Supervisor is to verify that preconditions for action execution are met before the execution is initiated. This primarily concerns the constraint that objects to be manipulated should be visible. This information is computed by the SPARK (Spatial Reasoning and Knowledge) module and made available to the system in ORO (the Open Robot Ontology) which provides central component of the Knowledge base of the system. See CHRIS.I [1] for details.

V. Experiments

A. Experiment 1- Completing Perception with Execution

In CHRIS.I we demonstrated a capability to learn to recognize actions including `take` and `put`. Here we first demonstrate how these action definitions can be completed with the execution component.

H: Put the toy on the left
R: I don't know how to put.
H: Grasp the toy.
R: Grasping the toy.
H: Release left
R: Releasing left
H: Finish learning.

Based on this learning we then demonstrated that the acquired execution knowledge could generalize to new instances of the action. We demonstrated that the robot correctly performed the command to put the box in the middle. This is illustrated in Fig 2A. In order to demonstrate that this knowledge could be exploited on a different robot, the learned definitions were shared via the SVN repository. Figure 2B illustrates the iCubGenoa01 using action definitions acquired in Lyon in order to perform the take and put actions.

B. Experiment 2- Learning New Actions

This experiment tests the ability of the system to learn new actions, both in terms of perception and execution. Here we focus on two actions which are `cover X with Y`, and `uncover X with Y`. We chose these actions as they will provide the basis for future work in shared planning for cooperation.

H: Cover the toy with the box.
R: I do not know how to cover.

H: Grasp the box.
R: Grasping the box.
H: Release the box on the toy.
R: Releasing the box on the toy.
H: Finish learning.

This dialog fragment illustrates how the system can acquire new sequences of action primitives in order to learn new composite actions. Here, "cover X with Y" is learned as the concatenation of grasp X and release X at Y. We demonstrated this same concatenative learning for the actions, put, take, cover and uncover. Note that put and cover have similar definitions, with reversed ordering of the arguments, demonstrating the flexibility of the argument binding capability.

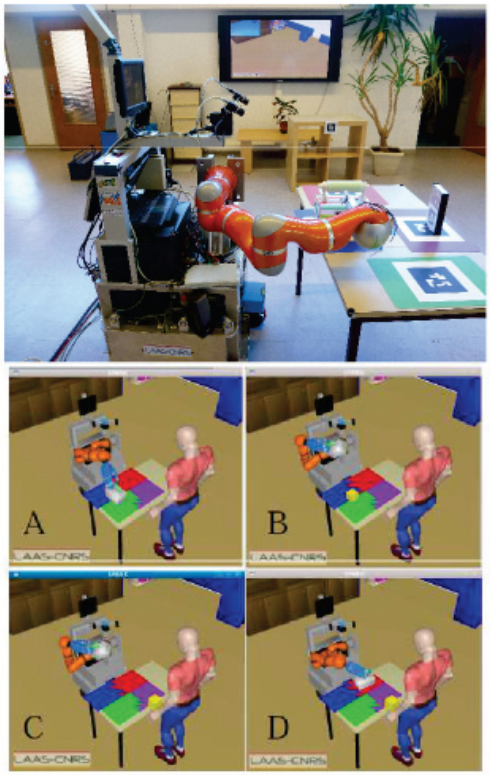


Figure 3: Above - Experimental platform Jido. The action of taking the box and putting it on the red-mat (cover X with Y) that was learned on iCubLyon01 was successfully executed in the Jido environment in Toulouse. A - B. Jido reaching for box and grasping. C – D. Jido puts box on red table mat.

C. *Experiment 3 – Cross platform generalization*

The Shared Plan Manager creates permanent definitions of these new actions, which can then be transferred via the SVN system for use on other robots at other sites. We could thus test the definition of `Cover X with Y` that was learned on the iCub in Lyon on the Jido robot in Toulouse.

Via the RAD Supervisor, the human asked Jido cover the red table-mat with the box (see Figure 3). The Supervisor retrieved the composite action definition, communicated the corresponding motor primitives corresponding to `grasp X` and `release X on Y` to Jido. Jido was thus able to produce the `cover X with Y` action, based on learning that had occurred on a morphologically distinct robot. Thus, despite this morphological difference, because of the abstraction at both perceptual and execution levels, action knowledge acquired on one platform can be exploited on another.

D. *Experiment 4 - Agency assignment with Kinect*

In behavior that involves object manipulation, the human hand has a special status as an agent. Indeed it has been shown that infants may prefer to assign agency to well known agents however they also rely on naïve physics and assign agency to objects that are moving on their own and in specific ways [4, 39]. In order to achieve accurate hand tracking we demonstrate here how the Kinect motion tracker can provide this capability. A module has been developed using the Kinect device in combination with OpenNI drivers[1] in order to track the user hands and add them to the Egosphere as standard objects. Since this module is on the platform specific side of the Egosphere, then no change is required to use its information. We achieved the same result using our standard vision system and visual markers on the human hand; however the approach with the Kinect is much more natural and robust. In the experiment the user was teaching system how to recognize *cover* and *uncover* and the system recognized these actions, and which hand performed them so it could describe it in the following way: "I detected that the *human hand* covered the *toy* with the *box*".

E. *Experiment 5 – Goal Directed Action Imitation*

This experiment, illustrated in detail in Figure 4, brings all of the functionality together. To arrive at this point, the robot should be able to both recognize and execute a set of actions. Here we demonstrate this with the `cover the toy with the box` action. This is illustrated briefly in Figure 2C and 2D. Figure 2C illustrates the human user showing the action to the robot. Figure 2D illustrates the robot now performing the recognized action. Full detail of

[1] Kinect is a hardware product by Microsoft (http://www.xbox.com/en-US/kinect). OpenNI.org release open source drivers for the Kinect device (http://openni.org/).

the experiment is provided in Figure 4. A video demonstrating this experiment is attached with the paper.

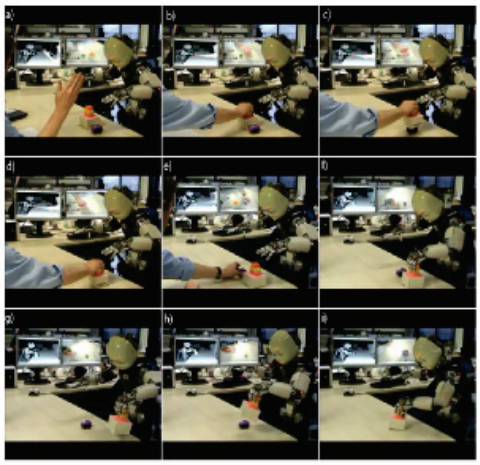


Figure 4: Experiment 5. Imitation. A. Calibration of hand recognition with Kinect. B-D. Human covers toy with box. E. Human repositions objects. F. Robot grasps box. G-I. Robot covers toy with box, completing the imitation.

VI. Discussion

Many of the mirroring skills demonstrated in the literature [40, 41] use the perceived motor state of the agent (i.e. its kinematic evolution over the action) to both recognize and execute actions. This has been combined with goal-based representations [10]. Our system is based on the fact that each action can be recognized by its perceptual consequences in changes in the world state (object states) and then performed by executing the associated motor commands. Those motor commands are not robot specific, but the primitives they call are, which implicitly solves the correspondence problem described in [8, 42]. Although we cannot argue that our system can cope with the same range of actions as a "trajectory based" systems, it is complimentary with such systems, and can be used at a higher level, for actions involving multiple arguments and symbolic goal achievement more than precise motor imitation. Indeed, this approach also emphasis the equifinal means of an action since the user can demonstrate an action and then the robot will achieve the same result with completely different trajectories.

Aspects of this work can thus be considered in the context of learning by imitation or demonstration, which is a major area of research in robot cognition today [8, 10, 40-42]. Our novel contributions to this domain include (1) the encoding of action in terms of perceptual state changes and composed motor primitives that can achieve these state changes, in a manner that allows the robot to learn new actions as perception – execution pairs, and then use this knowledge to perceive and imitate. (2) These actions can take several arguments, e.g. `AGENT put the OBJECT on the RECIPIENT`, which allows for the generalization of learned actions to entirely new contexts, with new objects and agents. This yields the equifinal component of action where the same goal can be achieved by different means. (3) We use spoken language interaction and visual perception to provide learning input to the system. In our long term research program, this provides that basis for learning to perform cooperative shared tasks purely through observation.

In our system actions are encoded using the effect they produce on the state of the world, the latter being abstracted in terms of unspecific quantities like relative position and orientation of objects and their visibility. The particular type of encoding we adopt for actions is therefore completely independent of the robot platforms, and can therefore be transferred between robots with different embodiments or perceptual systems. In previous work we showed how motor skills could be transferred between robots; this paper extends this work to action recognition and mirroring.

Our approach to action representation is consistent with and inspired by the 'teleological framework' [43, 44] that represents actions by relating three relevant aspects of reality (action, goal-state, and situational constraints) through the inferential 'principle of rational action', which assumes that: (a) the basic function of actions is to bring about future goal states; and that (b) agents will always perform the most efficient means action available to them within the constraints of the given situation. This approach is complimentary to existing approaches that take the "means" (e.g; aspects of demonstrated trajectories) into account [29, 36, 45]. Future research should consider how to combine these approaches.

VII. Acknowledgment

This research was supported by the European Commission under the Robotics and Cognitive Systems, ICT Project CHRIS (FP7-215805).

VIII. References

1. Lallée, S., et al. *Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception.* in *IROS*. 2010. Taipei.
2. Hommel, B., et al., *The theory of event coding (TEC): A framework for perception and action planning.* Behavioral and Brain Sciences, 2001. **24**(05): p. 849-878.
3. Rizzolatti, G. and L. Craighero, *The mirror-neuron system.* Annu. Rev. Neurosci., 2004. **27**: p. 169-192.
4. Király, I., et al., *The early origins of goal attribution in infancy.* Consciousness and Cognition, 2003. **12**(4): p. 752-769.
5. Dominey, P. and F. Warneken, *The basis of shared intentions in human and robot cognition.* New Ideas in Psychology, 2009: p. (in press).
6. Lallée, S., F. Warneken, and P. Dominey. *Learning to collaborate by observation.* in *Epirob*. 2009. Venice.

7. Csibra, G., et al., *Goal attribution without agency cues: the perception of [] pure reason'in infancy*. Cognition, 1999. **72**(3): p. 237-267.
8. Alissandrakis, A., C.L. Nehaniv, and K. Dautenhahn, *Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2002. **32**(4): p. 482-496.
9. Argall, B., et al., *A survey of robot learning from demonstration*. Robotics and Autonomous Systems, 2009. **57**(5): p. 469-483.
10.Calinon, S., F. Guenter, and A. Billard. *Goal-directed imitation in a humanoid robot*. in *ICRA*. 2005. Barcelona: IEEE.
11.Demiris, Y. and M. Johnson, *Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning*. Connection Science, 2003. **15**(4): p. 231-243.
12.Dillmann, R., *Teaching and learning of robot tasks via observation of human performance*. Robotics and Autonomous Systems, 2004. **47**(2-3): p. 109-116.
13.Woodward, A.L., *Infants selectively encode the goal object of an actor's reach*. Cognition, 1998. **69**(1): p. 1-34.
14.Woodward, A.L., *Infants' ability to distinguish between purposeful and non-purposeful behaviors*. Infant Behavior and Development, 1999. **22**(2): p. 145-160.
15.Woodward, A.L., J.A. Sommerville, and J.J. Guajardo, *How infants make sense of intentional action*. Intentions and intentionality: Foundations of social cognition, 2001: p. 149–169.
16.Kamewari, K., et al., *Six-and-a-half-month-old children positively attribute goals to human action and to humanoid-robot motion*. Cognitive Development, 2005. **20**(2): p. 303-320.
17.Mandler, J., ed. *Preverbal representation and language*. Language and space. 1996, MIT Press. 365–384.
18.Talmy, L., *Force dynamics in language and cognition*. Cognitive science, 1988. **12**(1): p. 49-100.
19.Kotovsky, L. and R. Baillargeon, *The development of calibration-based reasoning about collision events in young infants*. Cognition, 1998. **67**(3): p. 311-351.
20.Metta, G., et al. *The iCub humanoid robot: an open platform for research in embodied cognition*. in *PerMIS: Performance Metrics for Intelligent Systems Workshop*. 2008. Washington DC, USA.
21.Baillargeon Elizabeth, S., *Object permanence in five-month-old infants* 1*. Cognition, 1985. **20**(3): p. 191-208.
22.Mandler, J.M., *How to build a baby: II. Conceptual primitives*. PSYCHOLOGICAL REVIEW-NEW YORK-, 1992. **99**: p. 587-587.
23.Roy, D., *Semiotic schemas: A framework for grounding language in action and perception*. Artificial Intelligence, 2005. **167**(1-2): p. 170-205.
24.Siskind, J.M. *Visual event perception*. in *NEC Research Symposium*. 1998.
25.Spelke, E.S., et al., *Origins of knowledge*. PSYCHOLOGICAL REVIEW-NEW YORK-, 1992. **99**: p. 605-605.
26.Dominey, P. and J. Boucher, *Learning to talk about events from narrated video in a construction grammar framework*. Artificial Intelligence, 2005. **167**(1-2): p. 31-61.
27.Dominey, P. and J. Boucher, *Developmental stages of perception and language acquisition in a perceptually grounded robot*. Cognitive Systems Research, 2005. **6**: p. 243-259.
28.Mussa-Ivaldi, F.A., S.F. Giszter, and E. Bizzi, *Linear combinations of primitives in vertebrate motor control*. Proceedings of the National Academy of Sciences of the United States of America, 1994. **91**(16): p. 7534.
29.Mataric, M.J., et al. *Behavior-based primitives for articulated control*. in *Fifth international conference on simulation of adaptive behavior on From animals to animats 5*. 1998.
30.Williamson, M.M. *Postural primitives: Interactive behavior for a humanoid robot arm*. in *Fourth international conference on simulation of adaptive behavior on From animals to animats 4*. 1996.
31.Thomas, U., et al. *Error-tolerant execution of complex robot tasks based on skill primitives*. in *ICRA*. 2003. Taipei: IEEE.
32.Morrow, J.D. and P. Khosla. *Manipulation task primitives for composing robot skills*. in *ICRA*. 2002. Albuquerque: IEEE.
33.Sentis, L. and O. Khatib, *Synthesis of whole-body behaviors through hierarchical control of behavioral primitives*. International Journal of Humanoid Robotics, 2005. **2**(4): p. 505-518.
34.Firby, R.J. *Building symbolic primitives with continuous control routines*. in *First international conference on Artificial intelligence planning systems* 1992.
35.Paine, R.W. and J. Tani, *Motor primitive and sequence self-organization in a hierarchical recurrent neural network*. Neural Networks, 2004. **17**(8-9): p. 1291-1309.
36.Mussa-Ivaldi, F. and E. Bizzi, *Motor learning through the combination of primitives*. Philosophical Transactions of the Royal Society B: Biological Sciences, 2000. **355**(1404): p. 1755.
37.Lallée, S., et al., *Linking language with embodied teleological representations of action for humanoid cognition*. Frontiers in Neurobotics, 2010.
38.Sutton, S., et al. *Universal speech tools: The CSLU toolkit*. in *Fifth International Conference on Spoken Language Processing*. 1998.
39.Song, H., R. Baillargeon, and C. Fisher, *Can infants attribute to an agent a disposition to perform a particular action?* Cognition, 2005. **98**(2): p. B45-B55.
40.Johnson, M. and Y. Demiris. *Hierarchies of coupled inverse and forward models for abstraction in robot action planning, recognition and imitation*. in *AISB*. 2005.
41.Metta, G., et al., *Understanding mirror neurons: a bio-robotic approach*. Interaction studies, 2006. **7**(2): p. 197-232.
42.Nehaniv, C.L. and K. Dautenhahn, *2 The Correspondence Problem*. Imitation in animals and artifacts, 2002: p. 41.
43.Gergely, G. and G. Csibra, *Teleological reasoning in infancy: the naïve theory of rational action*. Trends in Cognitive Sciences, 2003. **7**(7): p. 287-292.
44.Gergely, G., *What should a robot learn from an infant? Mechanisms of action interpretation and observational learning in infancy*. Connection Science, 2003. **15**(4): p. 191-209.
45.Pattacini, U., et al. *An Experimental Evaluation of a Novel Minimum-Jerk Cartesian Controller for Humanoid Robots*. in *IROS*. 2010. Taipei.

Appendix 3

Linking Language with Embodied and Teleological Representations of Action for Humanoid Cognition

Title : Linking Language with Embodied and Teleological Representations of Action for Humanoid Cognition

Authors : Stephane Lallee, Carol Madden, Michel Hoen, Peter Ford Dominey

INSERM U846 Stem Cell and Brain Research Institute
Robot Cognition Laboratory
18, avenue du doyen Jean Lépine
69675 Bron cedex
France
Tél : +33-4 72 91 34 84
Fax : +33-4 72 91 34 61
{stephane.lallee, carol.madden, michel.hoen, peter.dominey}@inserm.fr

Abstract:

The current research extends our framework for embodied language and action comprehension to include a teleological representation that allows goal-based reasoning for novel actions. The objective of this work is to implement and demonstrate the advantages of a hybrid, embodied-teleological approach to action-language interaction, both from a theoretical perspective, and via results from human-robot interaction experiments with the iCub robot. We first demonstrate how a framework for embodied language comprehension allows the system to develop a baseline set of representations for understanding goal-directed actions such as "take", "cover", and "give". Spoken language and visual perception are input modes for these representations, and the generation of spoken language is the output mode. Moving towards a teleological (goal-based reasoning) approach, a crucial component of the new system is the representation of the subcomponents of these actions, which includes relations between initial enabling states, and final resulting states for these actions. We demonstrate how grammatical categories including causal connectives (e.g. because, if-then) can allow spoken language to enrich the learned set of state-action-state (SAS) representations. We then examine how this enriched SAS inventory enhances the robot's ability to understand perceived actions in which the environment inhibits goal achievement. The paper addresses how language comes to reflect the structure of action, and how it can subsequently be used as an input and output vector for embodied and teleological aspects of action.

1. Introduction – A framework for language and action

One of the central functions of language is to coordinate cooperative activity (Tomasello 2008). In this sense, much of language is about coordinating action. Indeed, as emphasized by Goldberg (1995, p. 5) "constructions involving basic argument structure are shown to be associated with dynamic scenes: experientially grounded gestalts, such as that of someone volitionally transferring something to someone else, someone causing something to move or change state …". Interestingly, this characterization is highly compatible with the embodied language comprehension framework, which holds that understanding language involves activation of experiential sensorimotor representations (Barsalou 1999, Bergen & Chang 2005, Fischer & Zwaan 2008, Zwaan & Madden 2005). We have pursued this approach in developing neurally inspired systems that make this link between language and action.

In this context, we first developed an action recognition system that extracted simple perceptual primitives from the visual scene, including contact or collision (Kotovsky & Baillargeon 1998), and composed these primitives into templates for recognizing events like give, take, touch and push. Siskind and collegues (Fern et al 2002) developed a related action learning capability in the context of force dynamics. A premise of this approach is that it is not so much the details of spatial trajectories of actions, but more their resulting states which characterize action in the context of perception and recognition (Bekkering et al. 2000). The resulting system provided predicate-argument representations of visually perceived events, which could then be used in order to learn the mapping between sentences and meaning. We demonstrated that naïve humans could narrate their actions which were perceived by the event recognition system, thus providing sentence-meaning inputs to the grammatical construction model, which was able to learn a set of grammatical constructions that could then be used to describe new instances of the same types of events (Dominey & Boucher 2005).

We subsequently extended the grammatical construction framework to robot action control. We demonstrated that the robot could learn new behaviors (e.g. Give me the *object*, where *object* could be any one of a number of objects that the robot could see) by exploiting grammatical constructions that define the mapping from sentences to predicate-argument representations of action commands. This work also began to extend the language-action framework to multiple-action sequences, corresponding to more complex behaviors involved in cooperative activity (Dominey, Mallet & Yoshida. 2009). Cooperation – a hallmark of human cognition (see Tomasello et al. 2005) – crucially involves the construction of action plans that specify the respective contribution of both agents, and the representation of this shared plan by both agents. Dominey and Warneken (2009) provided the Cooperator – a 6DEF arm and monocular vision robot - with this capability, and demonstrated that the resulting system could engage in cooperative activity, help the human, and perform role reversal, indicating indeed that it had a "bird's eye view" of the cooperative activity. More recently, Lallee et al. (2009) extended this work so that the robot could acquire shared plans by observing two humans perform a cooperative activity.

An important aspect of this research is that the source of meaning in language is derived directly from sensory-motor experience, consistent with embodied language processing theories (Barsalou 1999, Bergen & Chang 2005, Zwaan & Madden 2005). However, we also postulated that some aspects of language comprehension must rely on a form of "hybrid" system in which meaning might not be expanded completely into its sensory-motor manifestation (Madden et al. 2009). This would be particularly useful when performing goal-based inferencing and reasoning. Indeed, Hauser and Wood (2009) argue that understanding

action likely involves goal-based teleological reasoning processes that are distinct from the embodied simulation mechanisms for action perception. This is consistent with a hybrid approach to action understanding that we have recently proposed (Madden et al. 2009). In that model, action perception and execution take place in an embodied sensorimotor context, while certain aspects of planning of cooperative activities are implemented in an amodal system that does not rely on embodied simulation.

A fundamental limitation of this approach to date is that the system has no sense of the underlying goals for the individual or joint actions. This is related to the emphasis that we have placed on recognition and performance of actions, and shared action sequences, without deeply addressing the enabling and resulting states linked to these actions. In the current research, we extend our hybrid comprehension to address aspects of goal based reasoning, thus taking a first step towards the type of teleological reasoning advocated by Hauser and Wood (2009).

2. A new framework for action and language – combing teleological and embodied mechanisms

In Lallee et al. (2009) the iCub robot could observe two human agents perform a cooperative task, and then create a cooperative plan, which includes the interleaved temporal sequence of coordinated actions. It could then use that plan to take the role of either of the two agents in the learned cooperative task. This is illustrated in Figure 1. A limitation of this work is that the task is represented as a sequence of actions, but without explicit knowledge of the results of those actions, and the link between them. In the current work, this limitation is addressed by allowing the robot to learn for each action, what is the enabling state of the world which must hold for that action to be possible, and what is the resulting state that holds once the action has been performed. We will refer to this as the $S_E AS_R$ state-action-state representation of action. This is consistent with our knowledge that humans tend to represent actions in terms of goals – states that result from performance of the action (Woodward 1998).

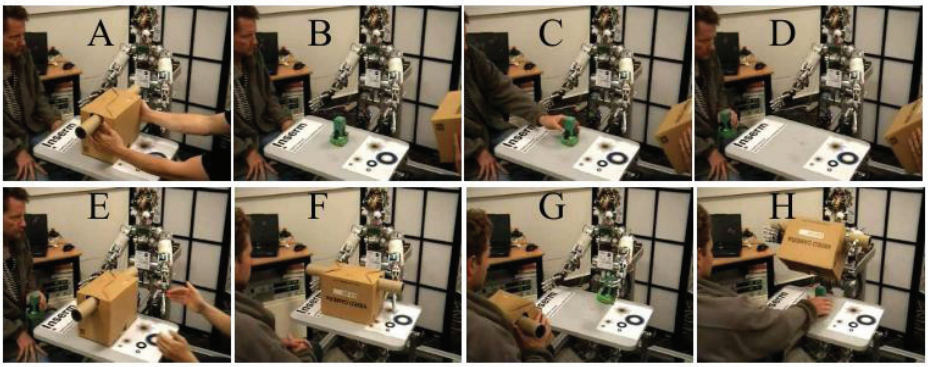


Figure 1. On-line learning of a cooperative task. A-B: Larry (left of robot) lifts the box that covers the toy. C-D: This allows Robert (right of robot) to take the toy. E: Larry replaces the box. F: Robot now participates. G: Human takes box, so Robot can take the toy. H: Robot takes box so human can take the toy.

Interestingly, we quickly encountered limitations of the perceptual system, in the sense that when an action causes an object to be occluded, the visual disappearance of that object is quite different from the physical disappearance of the object, yet both result in a visual disappearance. The ability to keep track of objects when they are hidden during a perceived action, and the more general notion of object constancy is one of the signatures of core object cognition (Spelke 1990, see Carey 2009). This introduces the notion that human cognition is built around a limited set of "core systems" for representing objects, actions, number and space (Spelke & Kinzler 2007). Robot cognition clearly provides a testing ground for debates in this domain, and the current study uses this platform to investigate the nature of the core system for agency. Embodied theories hold that actions are interpreted by mental simulation of the observed action, while teleological theories hold that this is not sufficient, and that a generative, rationality-based inferential process is also at work in action understanding (Gergely & Csibra 2003).

Event understanding often involves inferences of links between intentions, actions, and outcomes. Language can play an important role in helping children learn about relations between actions and their consequences (Bonawitz et al. 2009). This section provides an overview of how language is used to enrich perceptual representations of action, and some of the corresponding neurophysiological mechanisms that provide some of these capabilities. It is our belief that understanding these neurophysiological mechanisms can provide strong guidelines in constructing a system for robot event cognition in the context of human-robot cooperation.

2.1 Aspects of Language and Causality

One of the hallmarks of human cognition is the ability to understand goal-directed events. This ability surely entails the representation of events in terms of their causes and effects or goals (Bekkering et al. 2000, Sommerville and Woodward 2005), but how does it work? Although some theorists have postulated that causality itself is a conceptual primitive, it has become evident that causality can be decomposed into constituent elements (see Carey, 2009 for discussion). According to physicalist models of causality, causes and effects are understood in terms of transfer or exchange of physical quantities in the world, such as energy, momentum, impact forces, chemical and electrical forces (Talmy, 1988; Wolff, 2007), and nonphysical causation (e.g., forcing someone to decide) is understood by analogy to these physical forces. In this sense, physicalist models necessitate the ability to perceive kinematics, and dynamic forces, in order to represent causal relationships between entities. That is, to understand causality, one must have a body, and thus any implementation model of causal understanding necessitates an embodied system, to sense physical forces. *Dynamic* forces are invisible, such as the difference in the feeling of contact when an object is moving fast or slow, and how a pan feels when it is hot or cold. Because invisible dynamic forces map so well onto our experience of *kinematic* forces, or visual experience of forces (shape, size, position, direction, velocity, accelerations), humans often rely solely on visual information when attributing causal relationships in the world. In the same vein, causal understanding in non-human systems can be implemented through the use of kinematics as perceived via vision. Siskind (2002, Fern et al. 2001) has exploited the mapping of force dynamic properties into the visual domain, for primitives including contact, support and attachment. This results in robust systems in which event definitions are prespecified or learned, and then used for real-time event classification. Dominey & Boucher (2005)

employed a related method for the recognition of events including give, take, push, touch in the context of grounded language acquisition.

In the context of development, once a toddler is able to sense and understand physical forces in the environment, he has the tools to understand causal relationships. Pioneering studies have shown that this understanding of causality and causal language is acquired very early in development, as infants may already perceive cause-effect relationships at only 27 weeks (Leslie & Keeble, 1987), and toddlers can already express many types of causal language by the age of 2-3 years (Hood, Bloom, Brainerd, 1979; Bowerman, 1974). At this stage, exposure to language may help to accelerate the development of causal understanding. One study has shown that when toddlers are exposed to a causal relationship between 2 events accompanied by a causal description, they are more likely to initiate the first event to generate the second, and expect that the predictive relations will involve physical contact, compared to when they are exposed to the causal situation in the absence of causal language (Bonawitz et al 2009). That is, though the toddler associates the two events in either case, this association will not be used as a causal link unless this link is established explicitly via causal language such as "the block makes the light turn on".

In this way, language is used as a tool to further conceptual understanding of goal-directed events and actions by helping children integrate information about prediction, intervention, and contact causality. Thus, we can exploit language in our current system as a vector for establishing causal links between actions and their resulting states. In particular we are interested in the states that result from the "cover" and "give" actions which involve states related to the covered object being present, but invisible in the first case, and notions of change of possession in the second.

2.2 Cortical networks for language comprehension

In our effort to develop a system that can understand events and the state-transition relations between events, we can exploit knowledge of how language and event comprehension are implemented in the human nervous system. Language comprehension involves a cascade of computational operations starting from the decoding of speech in sensory areas to the emergence of embodied representations of the meaning of events corresponding to sensory-motor simulations (Barsalou 1999, Bergen & Chang 2005, Zwaan & Madden 2005, Rizzolatti & Fabbri-Destro 2009 for review). These representations are triggered via: observation of others engaged in sensory-motor events; imagination of events and the evocation of these experiences through language. Therefore, we consider the existence of two parallel but interacting systems: one system for language processing, ultimately feeding information processes into a second system, dedicated to the processing of sensory-motor events. These systems are highly interconnected and their parallel and cooperative work can ultimately bootstrap meaning representations. The second system will also accomodate the representation of elaborated events that implicates processes derived from a system sometimes referred to as a "social perception" network (Decety & Grezes, 2006; see Wible, Preus & Hashimoto, 2009 for review). This second network is directly involved in teleological aspects of reasoning, including agency judgments, attributing goals and intentions to agents, inferring rationality about ongoing events and predicting outcomes of the ongoing simulation (Hauser & Wood, 2009). We will present these two systems and show how they interact to form complex meaning representations through language comprehension.

One central view in the recent models of the cortical processing of language is that it occurs along two main pathways, mostly lateralized to the left cortical hemisphere (Hickok & Poeppel, 2007; Ullman, 2004). The first route is referred to as the ventral-stream. It is dedicated to the recognition of complex auditory (or visual) objects involving different locations along the temporal lobe and the ventralmost part of the prefrontal cortex (BA 45/46). The second one is named the dorsal-stream and is dedicated to the connection between the language system and the sensory-motor system, that is both implicated in the transformation of phonetic codes into speech gestures for speech production, but also in the temporal and structural decoding of complex sentences (Hoen et al., 2006; Meltzer et al., 2009). It implicates regions in the posterior part of the temporo-parietal junction, parietal and premotor regions and reaches the dorsal part of the prefrontal cortex (BA 44).

In the ventral pathway, speech sounds are decoded in or nearby primary auditory regions of the dorsal superior temporal gyrus (BA 41/42), before phonological codes can be retrieved from the middle posterior superior temporal sulcus (mp-STS - BA 22), and words recognized in regions located in the posterior middle temporal gyrus (pMTG – BA 22/37) (see Hickok & Poeppel, 2007 for review and Scott et al., 2006; Obleser et al., 2007). Then, these lexical symbols can trigger the reactivation of long-term stored sensory-motor experiences, either via implications of long-term autobiographic memory systems in the middle-temporal gyrus or in long-term sensory-motor memories, with a widespread storage inside the sensory-motor system. Therefore, complex meaning representation can actually engage locations from the ventral pathway but also memories stored inside the dorsal pathway (Hauk, Shtyrov & Pulvermüller, 2008; e.g. Tettamanti et al., 2005). This primary network feeds representation into a secondary-extended cortical network, whenever language leads to complex mental representations of complex events. Our initial computational models predicted dual structure-content pathway distinction (Dominey et al. 2003), which was subsequently confirmed in neuroimaging studies demonstrating the existence and functional implication of these two systems (Hoen et al., 2006), leading to further specification of the model (Dominey, Hoen and Inui, 2009).

2.3 Towards a Neurophysiological Model of Embodied and Teleological Event Comprehension

More recently, we extended this to a hybrid system in which sentence processing interacts both with a widespread embodied sensory-motor system, and with a more amodal system to account for complex event representation and scenario constructions operating on symbolic information (Madden, Hoen & Dominey, 2009). This second network, seems to engage bilateral parietal-prefrontal connections including bilateral activations in the parietal lobule for the perception and monitoring of event boundaries (Speer, Zacks & Reynolds, 2007) as well as dorsal prefrontal regions seemingly implicated in the global coherence monitoring of the ongoing mental representation elaboration (Mason & Just, 2006). The monitoring of complex event representation includes the ability of deciding if ongoing linguistic information can be inserted in the current representation and how it modifies the global meaning of this representation. These aspects rely on information and knowledge that are not primary characteristics of the language system per se but rather include general knowledge about causal relations between events, intentionality and agency judgments etc. These properties are sometimes called teleological reasoning and different authors have now shown that processes involving teleological reasoning are sustained by a distributed neural network, sometimes referred to as a "social perception" cognitive network that is closely related to the language

system (Wible, Preus & Hashimoto, 2009). This social perception network is implicated in teleological reasoning as determining agency or intentionality relations and involves regions as the right inferior parietal lobule (IP), the superior temporal sulcus (STS) and ventral premotor regions. All these regions are part of the well known mirror system (Decety and Grezes 2006). The TPJ or IP and STS regions, in addition to being part of the mirror system, are also heavily involved in other social cognition functions. Decety and Grezes, in an extensive review, (Decety and Grezes 2006) have designated the right temporo-parietal junction (TPJ) as the "social brain region. Theory of mind is the ability to attribute and represent other's mental states or beliefs and intentions or to "read their mind" ("predict the goal of the observed action and, thus, to "read" the intention of the acting individual" — from Decety and Grezes 2006). Therefore, it seems that regions that are implicated in social-cognition, that is to say regions implicated in agency, intentionality judgements on others are also implicated in the same judgements on a simulation / representation of mental simulations triggered by language.

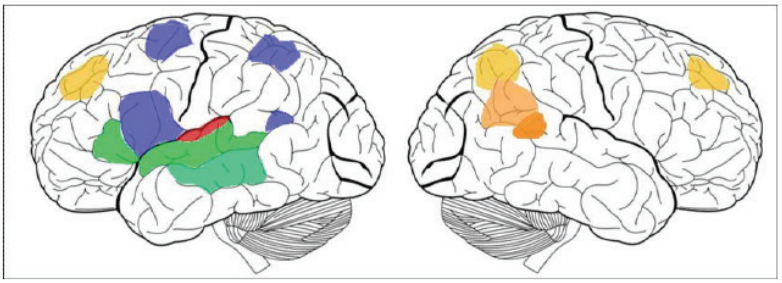


Figure 2. Cortical networks for language processing (Simplified). Ventral stream areas (Green) are part of a first network dedicated to speech decoding and phonological/lexical processing along the superior temporal sulcus (STS), middle temporal gyrus (MTG) and ventral prefrontal cortex (Pfc). Dorsal stream areas (Blue) constitute a sensory-motor interface implicated both in the transcription of phonological codes into articulatory codes (adapted from Hickock and Poeppel, 2007) but also in the temporal / structural organisation of complex sentence comprehension, and engage the left temporo-parietal junction, the parietal lobule and dorsal prefrontal regions (Hoen et al., 2006; Meltzer et al., 2009). The social perception or teleological cognition network (Oranges) is implicated in complex event representation and the attribution of agency, theory of mind in the right TPG (Orange, from Decety and Lamm, 2007), causality and intentionality in the posterior STS (Dark orange, from Saxe et al., 2004; Brass et al., 2007), and also comprises areas implicated in the global monitoring of the coherence of event representation (light Orange - from Mason & Just, 2006). Networks are shown in their specialized hemispheres but most contributions are bilateral.

Figure 2 illustrates a summary representation of the cortical areas involved in the hybrid, embodied-teleological model of language and event processing. The language circuit involves the frontal language system including BA44 and 45 with a link to embodied representations in the premotor areas, and in the more posterior parietal areas – both of which include mirror neuron activity in the context of action representation. This corresponds to the embodied component of the hybrid system. The teleological reasoning functions are implemented in a complimentary network that includes STS and TPJ/IP. In the current

research, while we do not model this hybrid system directly in terms of neural networks, we directly incorporate this hybrid architecture into the cognitive system for the robot.

3. Material and Methods

This section will present in three parts the physical platform, the behavioral scenarios, and the system architecture.

3.1 The iCub Humanoid and System Infrastructure

The current research is performed with the iCub, a humanoid robot developed as part of the RobotCub project (Tsagarakis et al. 2007). The iCub is approximately 1m tall, roughly the size and shape of a three-year-old child,, and its kinematic structure has a total of 53 degrees of freedom controlled by electric motors, primarily located in the upper torso. The robot hands are extremely dexterous and allow manipulation of objects thanks to their 18 degrees of freedom in total. The robot head is equipped with cameras, microphones, gyroscopes & linear accelerometers. The iCub is illustrated in Figures 1 and 4.
Spoken language processing and overall system coordination is implemented in the CSLU Rad toolkit. The system is provided with an "innate" recognition vocabulary including a set of action names (give, take, touch, cover, uncover), derived predicates (on, has), object names (block, star, sign), and causal language connectives (if-then, because). Vision is provided by a template-matching system (Spikenet^{TM}). State and action management are developed in C#. Interprocess communication is realized via the yarp protocol.

3.2 Experimental Scenarios

In this section we describe the experimental human-robot interaction scenarios that define the functional requirements for the system. The current scenarios concentrate on action understanding in the embodied and teleological frameworks. They demonstrate how language can be used (1) to enrich the representation of action and its consequences, and (2) to provide access to the structured representation of action definitions, and current knowledge of the robot.

3.2.1 Learning New Actions

In this scenario, the human performs physical actions with a set of visible objects in the robot's field of view. Typical actions include covering (and uncovering) one object with another, putting one object next to another, and briefly touching one object with another. For actions that the robot has not seen before, the robot should ask the human to describe the action. The robot should learn the action description (e.g. "The block covered the star"), and be capable of generalizing this knowledge to examples of the same action performed on different objects. For learned actions, the robot should be able to report on what it has seen. This should take place in a real-time, on-line manner. Knowledge thus acquired should be available for future use.

3.2.2 Learning the non-perceptual consequences of actions on objects

The causal relations between actions and the resulting states are not always trivial. When one object covers another, the second object "disappears" but is still physically present, beneath the covering object. In this scenario actions are performed that cause state changes, in terms

of the appearance and disappearance of objects. The robot should detect these changes and attempt to determine their cause. The cause may be known, based on prior experience. If not, then the robot should ask the human for clarification.

3.2.3 Use of Causal Constructions to Interrogate S_EAS_R representations

The links between actions and their enabling and resulting states correspond directly to grammatical expressions with the if-then construction. The sentence "If you want to take the block then the block must be visible" expresses an enabling relation, where the state "block visible" enables the action "take the block". In contrast, the sentence "If you cover the star with the block, then the star is under the block", or "If you cover the star with the block then the star is not visible" expresses a causal relation. This scenario should demonstrate how by using these forms of grammatical constructions, we can interrogate the system related to these enabling and causal relations.

3.2.4 Transfer of causal knowledge to new situations.

Here we want to demonstrate that if the robot learns about new action relations in one context then it can use this knowledge in another context. Concretely, in the cooperative task where Larry uncovers the toy so that Robot can pick it up, the robot should be able to begin to make the link between the resulting state of the "uncover" action as the enabling state of the subsequent "take" action. In this experiment, through a process of interrogation we will demonstrate that the robot has the knowledge necessary to form a plan for getting access to a covered object, by linking goals with resulting states of actions, and then establishing the enabling state as a new goal.

3.2.5 Extended usage

The goal of this experiment is to analyse the performance of the system under extended use, in order to observe the evolution of the KnowledgeBase, and the recognition capabilities of the system. We start with a naïve system (i.e. an empty KnowledgeBase), and then for the five actions *cover, uncover, give, take*, and *touch*, we expose the robot to each action with the block and the sign, and then in the transfer condition test the ability to recognize these actions with a new configuration (i.e. with the block and the star). We repeat this exhaustive exposure five times, in Phases 1 – 5. The dependant measure will be the number of presentations required for the five actions to be recognized in the training configuration, and transfer configuration, in each of the 5 phases.

3.3 Cognitive System Architecture

We developed a cognitive system architecture to respond to the requirements implied in Section 3.2, guided by knowledge of the cognitive linguistic mechanism in humans and their functional neurophysiology, and by our previous work in this area. The resulting system is not neuro-mimetic but its architecture is consistent with and inspired by our knowledge of the corresponding human system. We describe the architecture in the context of processing a new action, as in 3.2.1, and illustrated in Figure 4.

The human picks up the block and places it on the sign. Vision provides the front end of the perceptual system. Video data from the eyes of the iCub are processed by the Spikenet vision software which provides robust recognition for pretrained templates that recognize all objects

in the scene. Each template is associated with a name and the camera coordinates of the recognized location. One to 4 templates were required per object.

Based on our previous work, inspired by human developmental studies, we identified three perceptual primitives to be extracted from the object recognition, which would form the basis for generic action recognition – these are *visible(object, true/false)*, *moving(object, true/false)*, and *contact(obj1,obj2, true/false)*. These primitives are easily extracted from the Spikenet output based on position and its first derivative, and are provided as input to Temporal Segmentation. The temporal segmentation function returns the most recent set of segmented primitives that occurred within specified time window. This corresponds to our hypothesis that a given complex action will be constituted by a pattern of primitives that occur in a limited time window, separated in time by periods with no action. The resulting pattern of primitives for contact is illustrated in Figure 4C.

When the robot detects changes in the visual scene, the above processing is initiated. The Action Management function matches the resulting segmented perceptual primitives with currently defined action in the Knowledge Base. Each action in the Knowledge Base is defined by its pattern of action primitives, its name, the arguments it takes, any preconditions (i.e. the enabling state S_E in the S_EAS_R representation), and the resulting state. Thus, during action recognition, the Action Management function compares this set of segmented primitives with existing action patterns in the Knowledge Base. If no match is found then the system prompts the human to specify the action and its arguments, e.g. "I cover the sign with the block".

The State Management determines that as a result of the action, the World State has changed, and interrogates the user about this. The user then has the opportunity to describe any new relations that result from this action but that are not directly perceptible. When the block covers the sign, the sign is no longer visible, but still present. The State Management asks "Why is the sign no longer visible?" Thus the human can explain this loss of vision by saying "Because the block is on the sign." The action manager binds this relation in a generic way (i.e. it generalizes to new objects when the event "cover" is perceived) to the definition of "cover" (see Figure 4D).

If a match is found, then the system maps the concrete arguments in the current action segment with the abstract arguments in the action pattern. It can then describe what happened. For a recognized action, State Management updates the world state with any resulting states associated with that action. In the case of Cover, this includes encoding of the derived predicate on(block, star).

.

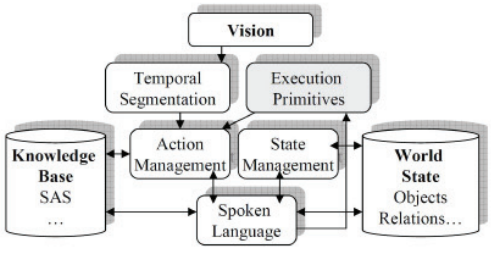


Figure 3. Cognitive System Architecture. See text for description.

4. Results:
4.1 & 4.2 Learning new actions and their derived consequences

Here we present results from an interaction scenario in which the user teaches the robot 4 new actions: Cover, uncover, give and take. In order to explain the system level functionality, details for learning are illustrated in Figure 4 for the action "Cover". The corresponding dialog is presented in Table 1.

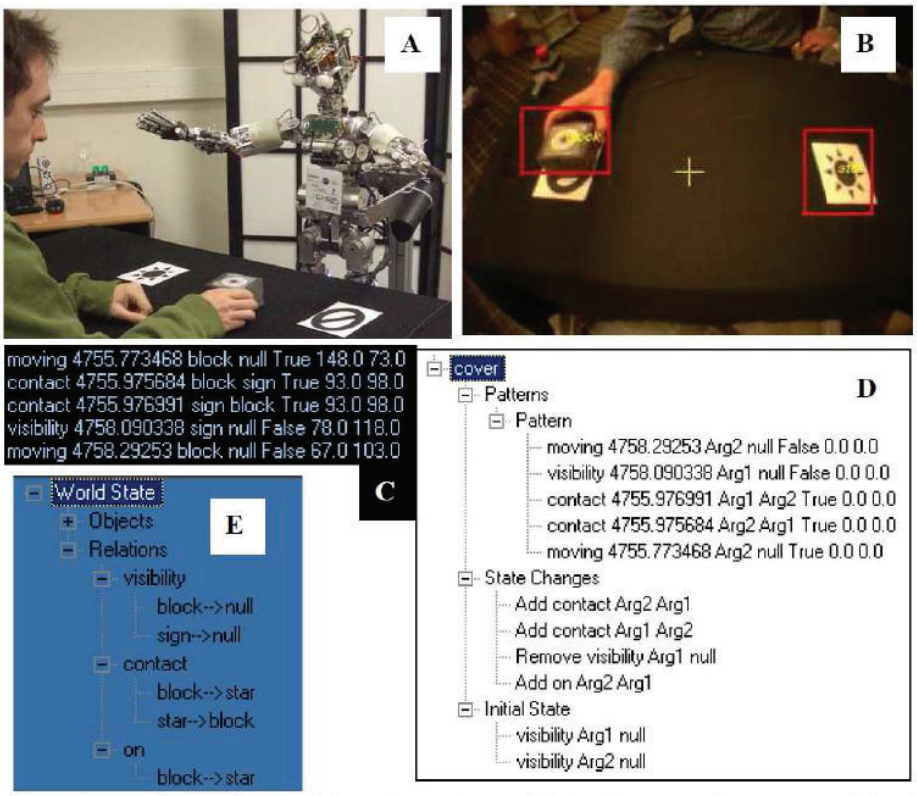


Figure 4. Learning and generalizing "Cover Arg1 with Arg2". A. Robot setup and visual scene before the action. B. Vision: Robot's view of scene after the block is put on the sign. C. Temporal Segmentation: Time ordered sequence of perceptual events observed during the action. D. Knowledge Base: Abstract pattern template for cover, along with the resulting state changes, and required initial state. E. Relations encoded in the World State after the "cover" is recognized in a new action "covers the star with the block".

For new actions (that have not yet been defined in KB) the system uses the set of observed primitives from Temporal Segmentation to generate a generic pattern of primitives to define the action (Figure 4C). If any unexpected perceptual changes occur, the system asks the human why this is the case, and the human can respond by describing any new relation that holds. For example, when the block covers the sign, the sign becomes not visible. The system asks the human why, and the human responds that this is "because the block is on the

sign". This new relation on(block, sign) is added as part of the generic definition of the cover action, illustrated in Figure 4D.

Learning "Cover" (0:42)
12:22:22 User : watch <user then places the block on the sign>
12:22:31 Robot : Need to Learn this one
12:22:58 User : cover the sign with the block
12:23:00 Robot : Did you say cover sign block ?
12:23:02 User : yes
12:23:04 Robot : Adding cover sign block
Learning Resulting Predicate "on" (0:17)
12:23:08 Robot : Why are the sign and the block close one each other ?
12:23:12 User : because the block is on the sign
12:23:15 Robot : Did you say because block on sign ?
12:23:17 User : yes
12:23:25 Robot : What should I do now ?
Transfer "Cover" to new situation (0:16)
12:24:07 Robot : What should I do now ?
12:24:09 User : watch
12:24:21 Robot : Recognized cover star block
12:24:23 Robot : What should I do now ?
User can interrogate the World State:
12:24:26 User : Tell about star
12:24:28 Robot : I will query info about the star
12:24:30 Robot : star contact block
12:24:33 Robot : block on star

Table 1. Interaction Fragment as Robot Learns Meaning of "Cover". Completion time (seconds).

Table 1 provides a record of the interaction in which the robot learns the meaning of "cover" and then displays this knowledge by recognizing cover in a new example. We observed that executing a given action like cover may sometimes lead to a different ordering of the segmented primitive events, e.g. detecting of the end of the block's movement may occur before or after the sign being visually obstructed. This is accommodated by encoding multiple patterns for a give action in the database. This redundant coding captures the physical redundancy that is expressed in the observations made by the system. The result is that when any of the appropriate patterns for an action are recognized, the action is recognized.

Enabling State	Action	Resulting State
Visible Arg1 Visible Arg2	Cover Arg1 with Arg2	Contact Arg1 Arg2 Visible Arg2 On Arg2 Arg1
Visible Arg2 True On Arg2 Arg1	Uncover Arg1 with Arg2	Visible Arg1 Visible Arg2
Visible Arg1 Visible Arg2	Give Arg1 to Arg2	Contact Arg1 Arg2 Visible Arg1 Visible Arg2 Has Arg2 Arg1
Contact Arg1 Arg2 Visible Arg1 Visible Arg2 On Arg2 Arg1	Take Arg1 from Arg2	Visible Arg1 Visible Arg2

Table 2: Enabling State – Action – Resulting State ($S_E A S_R$) representation for the learned actions.

A total of 5 distinct actions were learned and validated in this manner. The resulting definitions are summarized in Table 2. Figure 5 provides some performance statistics for learning these actions and then using the learned definitions to recognize new actions.

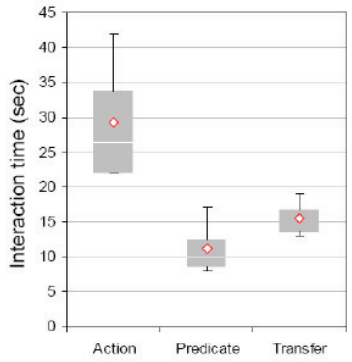


Figure 5. Performance effects for learning and transfer. Interaction times were recorded for the learning of four novel actions, learning the derived predicates for those actions, and then using the resulting definitions to recognize new instances of the four actions (Action, Predicate, and Transfer, respectively). Recognition in transfer to new actions (15.5 sec) was significantly faster than initial action learning time (29.25) $p < 0.05$..

4.2.3 Use of Causal Constructions to Interrogate S_EAS_R representations

This experiment demonstrates how the "if-then" construction can be used to extract the link between actions, the required enabling states, and the resulting states. Results are presented in Table 3.

```
13:43:10 User : What are the conditions for cover the sign with the block?
13:43:12 Robot : If you want to cover the block with the sign then
13:43:14 Robot : sign need to be visible
13:43:16 Robot : block need to be visible

13:43:23 User : What are the consequences of cover the sign with the block?
13:43:26 Robot : If you cover the sign with the block then
13:43:28 Robot : block will be contact with sign
13:43:30 Robot : sign will be contact with block
13:43:32 Robot : sign will no longer be visible
13:43:34 Robot : block will be on sign

The block is currently on the sign.
13:43:40 User : How can I make the sign visible?
13:43:43 Robot : If you want to make the sign visible then
13:43:45 Robot : You can uncover the sign
```

Table 3. Interaction fragment as robot explains the preconditions and results of the cover action.

4.2.4 Use of causal knowledge in teleological reasoning.

Here we consider a scenario similar to "uncover the block" scenario introduced in Section 1, and Figure 1. In this context, an object is covered by another, and the user's goal is to use the first object in a new task. The goal then is to find out how to gain access to the first object that is currently covered. The robot observes one human put the toy on the table, and another human cover the toy with the box. The objective is to begin to perform teleological reasoning about action sequences that have never been observed.

```
18:03:50 Robot : Recognized  cover star with block
18:04:05 User : What are the conditions to cover the sign with star
18:04:07 Robot : If you want to cover the sign with the star then
18:04:09 Robot : sign needs to be visible
18:04:11 Robot : star needs to be visible

The sign is visible, but the star is covered with the block, so the user
asks the robot how to make the sign visible:

18:04:18 User : How can I see the star
18:04:23 Robot : You can uncover the star

The user can now uncover the star by removing the block, and then achieve
the final goal which is to cover the sign with the star
```

Table 4. Interaction fragment where the robot performs basic goal-based reasoning about how to solve the task of getting the star when it is covered by the block, and then uses it to cover the sign.

This experiment demonstrates how the State-Action-State (S_EAS_R) representation provides the required information for goal-based reasoning.

4.2.5 Usage Study

We performed 6 additional experiments, which involved processing of 111 separate actions, to begin to evaluate the robustness of the system. Experiments 1-4 each started with an empty Knowledge Base, and examined the abililty to learn the 5 actions, and then transfer this knowledge to new object configurations. The key performance indices are (1) how many trials are required to learn an action with one set of objects, and (2) how well does this learning transfer to recognition of the same actions with different objects? Over the four experiments, a given action required 1.35 demonstrations to be learned accurately. This learning then transferred to new conditions on 70% of the new trials. 30% of the trials required additional learning before the actions were recognized under new conditions. Thus there is a significant effect of training on the ability to recognize new instances of learned action patterns ($p < 0.05$). Closer investigation revealed that in Exp 2 the vision system was generating false movement recognition which lead to a number of irrelevant patterns being learned. When only Experiment 1, 3 and 4 are considered, an average of 1.13 trials are required for learning, and the knowledge transfers to 100% of the new trials with no additional learning.

In Experiments 5 and 6 we retained the Knowledge Base from Experiment 4, and then tested it with a new user, and examined the evolution over two complete tests with the 5 actions and the two object configurations. In Experiment 5, a total of 6 additional demonstrations were

required to recognize the 5 actions in the two different object configurations. In Experiment 6 only 1 additional demonstration was required during the recognition of the 10 distinct actions. Overall these tests indicate that when the vision system is properly calibrated, the system is quite robust in the ability to learn generalized action recognition patterns.

5. Discussion

Part of the stated objective of this work has been to implement, and demonstrate the advantages of, a hybrid embodied-teleological approach to action-language interaction, both from a theoretical perspective and via results from human-robot interaction experiments with the iCub robot. This objective was motivated by our observation that true cooperation requires not only that the robot can learn shared action sequences, but that it represents how those actions are linked in a chain of state changes that lead to the goal. This means that the robot must be able to represent actions in terms of the states that allow them to be performed, the states that result from their performance including the "unseen" predicates, for example, related to object permanence.

We developed a perceptual system that extracts patterns of spatio-temporal visual properties in order to encode actions in terms of these patterns. We re-discovered that action and meaning are not purely perceptual (Carey 2009), and that additional properties related to object permanence and physical possession also form part of the meaning of action. Based on studies indicating that language can be used by toddlers to accelerate the acquisition of such knowledge (Bonawitz et al. 2009), when our cognitive system encounters unexpected results from an action, it interrogates the user, much like a developing child (Hood et al. 1979). This allows the user to explain, for example, that when the block covers the star, the star is not visible (but still there) *because* the block is *on* the star. We refer to these additional predicates (*on*, *has*) as derived predicates. This demonstrates that language can play an essential role refining the understanding of the meaning of action which is first approximated purely from the perceptual stream, by introducing derived predicates that become part of the meaning of the action. These predicates are encoded in the state changes that are to be introduced whenever the action is recognized. Thus, when the *give* and *take* actions are recognized, the derived predicate *has* (indicating possession) will be appropriately updated. We believe that this is a fundamental development in the link between language and action, because it goes beyond a purely identity mapping between sentences and meaning, and instead uses language to change and enrich forever the meaning of action as part of a developmental/learning process.

A crucial component of the new system is the representation of actions which includes the link to initial enabling states, and final resulting states. The resulting system produces a Knowledge Base that encodes the representation of action meanings, and a World State that encodes the current state of the world. As mentioned above, we demonstrate how grammatical constructions that exploit causal connectives (e.g. because) can allow spoken language to enrich the learned set of state-action-state (SAS) representations, by inserting derived predicates into the action definition. We also demonstrated how the causal connective "if – then" can be employed by the robot to inform the user about the links between enabling states and actions, and between actions and resulting states. Again, this extends the language – action interface beyond veridical action descriptions (or commands) to transmit more subtle knowledge about enabling and resulting states of actions, how to reach goals etc.

Indeed, in the context of the "hybrid" embodied and teleological system, we demonstrated how representations of enabling and resulting states provides the system with the knowledge necessary to make the link between goals as the resulting states of actions, and the intervening actions that are required. This is part of the basis of a teleological reasoning capability (Csibra 2003). In the current system, we have not implemented a full blown reasoning capability, that can perform forward and backward chaining on the states and action representations. This is part of our ongoing research.

In Foundations of Language, Jackendoff (2002) indicates that while languages may vary in their surface structure, the organization of the conceptual structure that they express appears more universal. We extended this notion to consider that indeed, the compositional structure of syntax is derived from that of the conceptual system (Dominey 2003), and Jackendoff agreed (Jackendoff 2003). In this context, one of the most promising results of the current research is the continued observation that language reflects the structure of conceptual representations. We have previously demonstrated this in situations where multiple actions are linked by shared states, resulting in descriptions such as "Larry took the toy that Robert uncovered with the box" (Dominey & Boucher 2005). The current work extends this to include functional and causal links between elements in the SAS representations (e.g. the if-then constructions in Tables 3 and 4).

We are currently working to integrate this S_EAS_R framework into our existing cooperative action framework (Dominey, Mallet & Yoshida 2009, Dominey & Warneken 2009). We will first demonstrate that the mechanism presented here for learning the perceptual patterns associated with perceived actions can be applied to learning motor patterns associated with executed actions. This will result in further enriched action representations that include the enabling and resulting states, the perceptual primitive patterns, and the action primitive patterns. We will then use these representations in the context of learning cooperative tasks by observation. This will yield a situation in which the robot can represent the trajectory from initial state to final goal state via coordinated action sequence, and will thus provide the basis for intentional reasoning, and the extension of the teleological reasoning to cooperative activity.

Acknowledgements:
This research is supported by the European Commission FP7 ICT Projects CHRIS and Organic, and French ANR Projects Amorces and Comprendre.

References:

Barsalou LW (1999) Perceptual symbol systems, Behavioral and Brain Sciences, 22, 577-660

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

Bergen B, Chang N (2005) Embodied Construction Grammar in Simulation-Based Language Understanding. In press. J.-O. Östman and M. Fried (eds.). *Construction Grammar(s): Cognitive and Cross-Language Dimensions*. Johns Benjamins

Bonawitz, E.B., Horowitz, A., Ferranti, D., Schulz, L. (2009) The Block Makes It Go: Causal Language Helps Toddlers Integrate Prediction, Action, and Expectations about Contact Relations. Proceedings of the Thirty-first Cognitive Science Society.

Bowerman, M. (1974). Learning the structure of causative verbs: A study in the relationship of cognitive, semantic, and syntactic development. Proceedings of Research in Child Language Development, 8, 142-178.

Csibra G (2003) Teleological and referential understanding of action in infancy, *Phil. Trans. R. Soc. Lond.* B (2003) **358**, 447–458

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

Decety, J. & Grèzes, J. (2006). The power of simulation: Imagining one's own and other's behavior. *Brain Research, 1079*, 4-14.

Decety, J., Lamm, C. (2007). The role of the right temporoparietal junction in social interaction: how low-level computational processes contribute to meta-cognition. Neuroscientist, 13:580-93.

Dominey PF (2003) A conceptuocentric shift in the characterization of language. Comment on Jackendoff, BBS, 674-675.

Dominey PF, Hoen M, Lelekov T, Blanc JM (2003) Neurological basis of language in sequential cognition: Evidence from simulation, aphasia and ERP studies, Brain and Language, Brain and Language, 86(2):207-25

Dominey , P.F., Warneken, F. (2009) The origin of shared intentions in human-robot cooperation, *New Issues in Psychology*

Dominey PF, Inui T, Hoen M. (2009) Neural network processing of natural language: II. Towards a unified model of corticostriatal function in learning sentence comprehension and non-linguistic sequencing. Brain Lang. May-Jun;109(2-3):80-92.

Dominey, P.F. and J.D. Boucher, Learning to talk about events from narrated video in a construction grammar framework. *Artificial Intelligence*, 2005. 167(1-2): p. 31-61.

Dominey, P.F., Mallet, A., Yoshida, E. (2009) Real-Time Spoken-Language Programming For Cooperative Interaction With A Humanoid Apprentice, *Intl. J. Humanoid Robotics*, 6(2) 147-171

Fern A, Givan R, Siskind JM (2002) Specific-to-General Learning for Temporal Events with Application to Learning Event Definitions from Video, Journal of Aritificial Intelli-gence Research, 17, 379-449

Fischer, M., & Zwaan, R. A. (2008). Embodied language: A review of the role of the motor system in language comprehension. Quarterly Journal of Experimental Psychology.

Gergely G, Csibra G (2003) Teleological reasoning in infancy: the naïve theory of rational action, Trends in Cognitive Science, 7(7) 287-292

Goldberg, A. (1995) Constructions: A construction grammar approach to argument structure. University of Chicago Press.

Hauser, M. and J. Wood, Evolving the Capacity to Understand Actions, Intentions, and Goals. *Annu Rev Psychol*, 2009.

Hickok G, Poeppel D. (2007). The cortical organization of speech processing. Nat Rev Neurosci, 8:393-402.

Hoen M, Pachot-Clouard M, Segebarth C, Dominey PF. (2006). When Broca experiences the Janus syndrome: an ER-fMRI study comparing sentence comprehension and cognitive sequence processing. Cortex, 42:605-23.

Hood, L., Bloom, L., & Brainerd, C.J. (1979). What, When, and How about Why: A Longitudinal Study of Early Expressions of Causality. Monographs of the Society for Research in Child Development, 44, 1-47.

Jackendoff R (2002) Foundations of Language: Brain, Meaning, Grammar, Evolution. Oxford University Press

Jackendoff R (2003) Précis of Foundations of Language: Brain, Meaning, Grammar, Evolution, BBS (26) 651-707.

Kotovsky L., Baillargeon R., (1998) The development of calibration-based reasoning about collision events in young infants, Cognition 67 311–351.

Kurby, C.A. & Zacks JM. Segmentation in the perception and memory of events. Trends Cogn Sci., 12:72-9.

Lallee, S., Warkeken F., Dominey, P.F. (2009) Learning to collaborate by observation and spoken language, *Ninth International Conference on Epigenetic Robotics,* Venice.

Leslie, A. M. & Keeble, S. (1987). Do six-month-old infants perceive causality? Cognition, 25, 265-288.

Madden, C., M. Hoen, and P.F. Dominey, A cognitive neuroscience perspective on embodied language for human-robot cooperation. *Brain and Language*, 2009.

Mason, R.A., & Just, M.A. (2006). Neuroimaging contributions to the understanding of discourse processes. In M. Traxler and M. A. Gernsbacher (Eds.), Handbook of Psycholinguistics (pp. 765-799). Amsterdam: Elsevier.

Meltzer, J.A., McArdle, J.J., Schafer, R.J., Braun, A.R. (2009). Neural Aspects of Sentence Comprehension: Syntactic Complexity, Reversibility, and Reanalysis. Cereb Cortex, In Press.

Michotte, A. (1963). *The Perception of Causality*. New York: Basic Books.

N. T. Sahin, S. Pinker, S. S. Cash, D. Schomer, E. Halgren, (2009). Science 326, 445.

Obleser, J., Zimmermann, J., Van Meter, J., Rauschecker, J.P. (2007). Multiple stages of auditory speech perception reflected in event-related FMRI. Cereb Cortex, 17(10):2251-7.

Pulvermüller, F., Shtyrov, Y., Hauk, O. (2009). Understanding in an instant: neurophysiological evidence for mechanistic language circuits in the brain. Brain Lang, 110:81-94.

Rizzolatti G, Fabbri-Destro M. (2009). Mirror neurons: from discovery to autism. Exp Brain Res. In press.

Sahin, N.T., Pinker, S., Cash, S.S., Schomer, D., Halgren, E. (2009). Science 326, 445.

Saxe, R., Xiao, D.K., Kovacs, G., Perrett, D.I., Kanwisher, N. (2004). A region of right posterior superior temporal sulcus responds to observed intentional actions. Neuropsychologia, 42:1435-46.

Scott, S.K., Rosen, S., Lang, H., Wise, R.J. (2006). Neural correlates of intelligibility in speech investigated with noise vocoded speech--a positron emission tomography study. J Acoust Soc Am, 120:1075-83.

Siskind, J.M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research, 15*, 31-90.

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

Speer, N.K., Zacks, J.M., Reynolds, J.R. (2007). Human brain activity time-locked to narrative event boundaries. Psychol Sci, 18:449-55.

Spelke ES, Kinzler KD (2007) Core knowledge, Developmental Science 10:1 pp 89–96

Spelke, E.S. (1990). Principles of object perception. Cognitive Science, 14, 29–56.

Talmy, L. (1988). Force dynamics in language and cognition. Cognitive Science, 12, 49-100.

Tettamanti, M., Buccino, G., Saccuman, M. C., Gallese, V., Danna, M., Scifo, P., et al. (2005). Listening to action-related sentences activates fronto-parietal motor circuits. Journal of Cognitive Neuroscience, 17, 273–281

Tsagarakis N.G., Metta G., Sandini G., Vernon D., Beira R., Becchi F., Righetti L., Victor J.S., Ijspeert A.J., Carrozza M.C. and Caldwell D.G.. (2007) iCub – The Design and Realization of an Open Humanoid Platform for Cognitive and Neuroscience Research. Advanced Robotics, Vol 21, No. 10

Ullman MT. (2004). Contributions of memory circuits to language: the declarative/procedural model. Cognition, 92:231-70.

Ullman, M.T. (2004). Contributions of memory circuits to language: the declarative/procedural model. Cognition, 92:231-70.

Wible, C. G.; Preus, A. P.; Hashimoto, R. (2009). A Cognitive Neuroscience View of Schizophrenic Symptoms: Abnormal Activation of a System for Social Perception and Communication. Brain Imaging Behav, 3(1): 85-110.

Wolff, P. (2007). Representing causation. Journal of Experimental Psychology: General, 136, 82-111.Barsalou, L.W., et al., Grounding conceptual knowledge in modality-specific systems. *Trends Cogn Sci*, 2003. 7(2): p. 84-91.

Woodward, A. L. (1998) Infants selectively encode the goal object of an actor's reach. Cognition 69:1–34.

Zwaan, R.A., & Madden, C.J. (2005). Embodied Sentence Comprehension. In: D. Pecher & R.A. Zwaan (Eds.), The Grounding of Cognition: The Role of Perception and Action in Memory, Language, and Thinking. Cambridge, UK: Cambridge University Press.

Appendix 4

The EFAA's OPC format specification

This document has been produced to define the OPC object properties formalization and share them among the project partners. It defines various keywords designing object properties that can be updated or queried by other modules.

IDs

Items are assigned a unique ID returned by the OPC when something is added to the database. ID is a property of an object and allows storing every object, even those which don't have any name. Any ID is unique within the OPC.

Entity

Entity defines the type of the item. Here are the different entities defined within the *efaaHelpers.h* header file:

```
#define EFAA_OPC_ENTITY_OBJECT              ("object")
#define EFAA_OPC_ENTITY_TABLE               ("rt_table")
#define EFAA_OPC_ENTITY_ROBOT               ("robot")
#define EFAA_OPC_ENTITY_CURSOR              ("cursor")
#define EFAA_OPC_ENTITY_MATH                ("math")
#define EFAA_OPC_ENTITY_LOCATION            ("location")
#define EFAA_OPC_ENTITY_EMO_ROBOT           ("emo_robot")
#define EFAA_OPC_ENTITY_EMO_HUMAN           ("emo_human")
#define EFAA_OPC_ENTITY_BODY_PART           ("body_part")
```

Spatial Properties

The position of an object in the robot reference frame is coded by 3 properties of double type. The unit is meter.

```
#define EFAA_OPC_OBJECT_ROBOTPOSX_TAG       ("robot_position_x")
#define EFAA_OPC_OBJECT_ROBOTPOSY_TAG       ("robot_position_y")
#define EFAA_OPC_OBJECT_ROBOTPOSZ_TAG       ("robot_position_z")
```

The dimensions of an object (bouding box) are coded by 3 properties of integer type. The unit is mm.

```
#define EFAA_OPC_OBJECT_RTDIMX_TAG          ("rt_dim_x")
#define EFAA_OPC_OBJECT_RTDIMY_TAG          ("rt_dim_y")
#define EFAA_OPC_OBJECT_RTDIMZ_TAG          ("rt_dim_z")
```

Affordances & PMP

For now the grasping configuration can be a property of an entity with the following tag & values. It can be used to send the proper command to PMP depending on the target.

```
#define EFAA_OPC_OBJECT_GRASPCONF_TAG       ("graspConfiguration")
#define EFAA_OPC_OBJECT_GRASPCONF_UP        ("up")
#define EFAA_OPC_OBJECT_GRASPCONF_SIDE      ("side")
#define EFAA_OPC_OBJECT_GRASPCONF_TOP       ("top")
```

Spatial relations

Spatial relations between objects are currently checked by the objRelationFinder module. It pushes properties of type list which store the list of IDs of objects.

e.g.: ((id 1) (contains (3 4 5)) (isContained (6))) states that object 1 contains objects 3, 4 and 5 and is contained within object 6.

```
#define EFAA_OPC_OBJECT_SPATIAL_CONTAINS    ("contains")
#define EFAA_OPC_OBJECT_SPATIAL_CONTAINED   ("isContained")
#define EFAA_OPC_OBJECT_SPATIAL_INTERSECTS  ("intersects")
```

The entity « body part » (human awareness)

The human detection modules will detect the humans (using the kinect for example), split it into part and push those parts into the OPC so higher level modules can use this information. A body part possesses a name, the «*robot_position_*...» property and an «*owner*» property. The «*owner*» property is of the string type and refers to the name of the human whom those parts belong to. The different body part names can be:

```
#define EFAA_OPC_BODY_PART_TYPE_HEAD            ("head")
#define EFAA_OPC_BODY_PART_TYPE_HAND_L          ("handLeft")
#define EFAA_OPC_BODY_PART_TYPE_HAND_R          ("handRight")
```

For example, to get the ID of Ilaria's face you can send the query:

```
ask ((entity==body_part)&&(name==head)&&(owner==Illaria)&&(isPresent==1))
```

iCubGUI

Every entity present in the OPC that has its spatial properties (*robot_position_x*, *robot_position_y*, *robot_position_z*) set and the tag (*isPresent* 1) will be displayed in the GUI. Module responsible for this display is objLocationTransformer. The color of an object within the GUI is coded by 4 properties of integer type value:

```
#define EFAA_OPC_OBJECT_GUI_COLOR_R         ("color_r")
#define EFAA_OPC_OBJECT_GUI_COLOR_G         ("color_g")
#define EFAA_OPC_OBJECT_GUI_COLOR_B         ("color_b")
#define EFAA_OPC_OBJECT_GUI_COLOR_ALPHA     ("color_aplha")
```

References

Ach, N. (1935). "Analyse des Willens." Handbuch der biologischen Arbeitsmethoden.

Aflalo, T. N. and M. S. A. Graziano (2006). "Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array." The Journal of neuroscience **26**(23): 6288-6297.

Alissandrakis, A., C. L. Nehaniv, et al. (2002). "Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **32**(4): 482-496.

Allen, J. and G. Ferguson (1994). "Actions and events in interval temporal logic." Journal of logic and computation **4**(5): 531.

Allen, J. F. (1984). "Towards a general theory of action and time." Artificial Intelligence **23**(2): 123-154.

Amunts, K., A. Schleicher, et al. (1999). "Broca's region revisited: cytoarchitecture and intersubject variability." The Journal of comparative neurology **412**(2): 319-341.

Anderson, M. L. (2003). "Embodied cognition: A field guide." Artificial Intelligence **149**(1): 91-130.

Arbib, M. A. (1998). Schema theory, MIT Press.

Arbib, M. A. (2005). "From monkey-like action recognition to human language: An evolutionary framework for neurolinguistics." Behavioral and Brain Sciences **28**(02): 105-124.

Argall, B., S. Chernova, et al. (2009). "A survey of robot learning from demonstration." Robotics and Autonomous Systems **57**(5): 469-483.

Aukrust, V. G. (1996). "Learning to talk and keep silent about everyday routines: A study of verbal interaction between young children and their caregivers." Scandinavian journal of educational research **40**: 311-324.

Baillie, J. C. (2005). Urbi: Towards a universal robotic low-level programming language, IEEE.

Baldwin, D., A. Andersson, et al. (2008). "Segmenting dynamic human action via statistical structure." Cognition **106**(3): 1382-1407.

Barsalou, L. W. (2008). "Grounded cognition." Annu. Rev. Psychol. **59**: 617-645.

Becchio, C. and C. Bertone (2002). "Il problema della condivisione." Sistemi intelligenti **14**(2): 207-216.

Becchio, C. and C. Bertone (2004). "Wittgenstein running: Neural mechanisms of collective intentionality and we-mode." Consciousness and Cognition **13**(1): 123-133.

Becchio, C. and C. Bertone (2005). "Beyond Cartesian subjectivism: Neural correlates of shared intentionality." Journal of Consciousness Studies **12**(7): 20-30.

Benjamin, D. P., D. Lyons, et al. (2004). Designing a robot cognitive architecture with concurrency and active perception.

Bernstein, N. A. (1967). "The co-ordination and regulation of movements."

Bernstein, S. (1927). "Über ein geometrisches Theorem und seine Anwendung auf die partiellen Differentialgleichungen vom elliptischen Typus." Mathematische Zeitschrift **26**(1): 551-558.

Biggs, G. and B. MacDonald (2003). A survey of robot programming systems, Citeseer.

Bonath, B., T. Noesselt, et al. (2007). "Neural basis of the ventriloquist illusion." Current Biology **17**(19): 1697-1703.

Botvinick, M. and J. Cohen (1998). "Rubber hands' feel'touch that eyes see." Nature **391**(6669): 756-756.

Braitenberg, V. and A. Schüz (1998). Cortex: statistics and geometry of neuronal connectivity, Springer Berlin:.

Bratman, M. E. (1992). "Shared cooperative activity." The Philosophical Review **101**(2): 327-341.

Bratman, M. E. (1993). "Shared intention." Ethics **104**(1): 97-113.

Brodmann, K. (1909). "Vergleichende lokalisationslehre der grobhirnrinde." Barth, Leipzig.

Buccino, G., F. Binkofski, et al. (2001). "Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study." European Journal of Neuroscience **13**(2): 400-404.

Burghart, C., R. Mikut, et al. (2005). A cognitive architecture for a humanoid robot: A first approach, IEEE.

Burns, E., S. Razzaque, et al. (2005). The Hand is Slower than the Eye: A quantitative exploration of visual dominance over proprioception, IEEE.

Calinon, S., F. Guenter, et al. (2005). Goal-directed imitation in a humanoid robot. ICRA, Barcelona, IEEE.

Calvo-Merino, B., D. E. Glaser, et al. (2005). "Action observation and acquired motor skills: an fMRI study with expert dancers." Cerebral Cortex **15**(8): 1243-1249.

Calvo-Merino, B., J. Grèzes, et al. (2006). "Seeing or doing? Influence of visual and motor familiarity in action observation." Current Biology **16**(19): 1905-1910.

Cappe, C. and P. Barone (2005). "Heteromodal connections supporting multisensory integration at low levels of cortical processing in the monkey." European Journal of Neuroscience **22**(11): 2886-2902.

Carr, L., M. Iacoboni, et al. (2003). "Neural mechanisms of empathy in humans: a relay from neural systems for imitation to limbic areas." Proceedings of the National Academy of Sciences of the United States of America **100**(9): 5497.

Cassimatis, N. L., J. G. Trafton, et al. (2004). "Integrating cognition, perception and action through mental simulation in robots." Robotics and Autonomous Systems **49**(1): 13-23.

Chaminade, T. and J. Decety (2002). "Leader or follower? Involvement of the inferior parietal lobule in agency." NeuroReport **13**(15): 1975.

Chong, H. Q., A. H. Tan, et al. (2007). "Integrated cognitive architectures: a survey." Artificial Intelligence Review **28**(2): 103-130.

Cohen, P. R. and H. J. Levesque (1990). "Intention is choice with commitment." Artificial Intelligence **42**(2-3): 213-261.

Collins-Sussman, B., B. W. Fitzpatrick, et al. (2004). Version control with subversion, O'Reilly Media, Inc.

Cordes, D., V. M. Haughton, et al. (2000). "Mapping functionally related regions of brain with functional connectivity MR imaging." American Journal of Neuroradiology **21**(9): 1636-1644.

Csibra, G. (2008). "Goal attribution to inanimate agents by 6.5-month-old infants." Cognition **107**(2): 705-717.

Csibra, G., G. Gergely, et al. (1999). "Goal attribution without agency cues: the perception of [] pure reason'in infancy." Cognition **72**(3): 237-267.

Cytowic, R. E. (2002). Synesthesia: A union of the senses, The MIT Press.

Damasio, A. R. (1989). "The brain binds entities and events by multiregional activation from convergence zones." Neural Computation **1**(1): 123-132.

Damasio, A. R. and H. Damasio (1994). "Cortical systems for retrieval of concrete knowledge: The convergence zone framework." Large-scale neuronal theories of the brain: 61-74.

Decety, J., T. Chaminade, et al. (2002). "A PET exploration of the neural mechanisms involved in reciprocal imitation." Neuroimage **15**(1): 265-272.

Decety, J., J. Grezes, et al. (1997). "Brain activity during observation of actions. Influence of action content and subject's strategy." Brain **120**(10): 1763.

Decety, J. and J. A. Sommerville (2003). "Shared representations between self and other: a social cognitive neuroscience view." Trends in Cognitive Sciences **7**(12): 527-533.

Dekker, A. H. (1994). "Kohonen neural networks for optimal colour quantization." Network: Computation in Neural Systems **5**(3): 351-367.

Demiris, Y. and M. Johnson (2003). "Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning." Connection Science **15**(4): 231-243.

Descartes, R. and M. Moriarty (2008). Meditations on first philosophy: With selections from the objections and replies, Oxford University Press, USA.

Dillmann, R. (2004). "Teaching and learning of robot tasks via observation of human performance." Robotics and Autonomous Systems **47**(2-3): 109-116.

Dinstein, I., C. Thomas, et al. (2008). "A mirror up to nature." Current Biology **18**(1): R13-R18.

Dobelle, W. H. (2000). "Artificial vision for the blind by connecting a television camera to the visual cortex." ASAIO journal **46**(1): 3.

Dominey, P., M. Alvarez, et al. (2005). Robot command, interrogation and teaching via social interaction.

Dominey, P. and J. Boucher (2005). "Developmental stages of perception and language acquisition in a perceptually grounded robot." Cognitive Systems Research **6**: 243-259.

Dominey, P. and J. Boucher (2005). "Learning to talk about events from narrated video in a construction grammar framework." Artificial Intelligence **167**(1-2): 31-61.

Dominey, P., A. Mallet, et al. (2007). Progress in programming the hrp-2 humanoid using spoken language. IEEE International Conference on Robotics and Automation.

Dominey, P., A. Mallet, et al. (2007). Real-time cooperative behavior acquisition by a humanoid apprentice. International Conference on Humanoid Robotics, Pittsburg, Pennsylvania.

Dominey, P. and F. Warneken (2009). "The basis of shared intentions in human and robot cognition." New Ideas in Psychology: (in press).

Eckerman, C. O. (1993). "Toddlers' Achievement of Coordinated Action with Conspecincs: A Dynamic Systems Perspective."

Ehrsson, H. H., N. P. Holmes, et al. (2005). "Touching a rubber hand: feeling of body ownership is associated with activity in multisensory brain areas." The Journal of neuroscience **25**(45): 10564-10573.

Ehrsson, H. H., C. Spence, et al. (2004). "That's my hand! Activity in premotor cortex reflects feeling of ownership of a limb." Science **305**(5685): 875.

Engel, S. A., G. H. Glover, et al. (1997). "Retinotopic organization in human visual cortex and the spatial precision of functional MRI." Cerebral Cortex **7**(2): 181.

Faillenot, I., I. Toni, et al. (1997). "Visual pathways for object-oriented action and object recognition: functional anatomy with PET." Cerebral Cortex **7**(1): 77.

Farrer, C., N. Franck, et al. (2003). "Modulating the experience of agency: a positron emission tomography study." Neuroimage **18**(2): 324-333.

Felleman, D. J. and D. C. Van Essen (1991). "Distributed hierarchical processing in the primate cerebral cortex." Cerebral Cortex **1**(1): 1.

Fiehler, K. and F. Rösler (2010). "Plasticity of multisensory dorsal stream functions: Evidence from congenitally blind and sighted adults." Restorative Neurology and Neuroscience **28**(2): 193-205.

Firby, R. J. (1992). Building symbolic primitives with continuous control routines. First international conference on Artificial intelligence planning systems

Fitzpatrick, P., G. Metta, et al. (2007). "Towards Long-Lived Robot Genes." Robotics and Autonomous Systems **56**(1): 29-45.

Flash, T. and B. Hochner (2005). "Motor primitives in vertebrates and invertebrates." Current opinion in neurobiology **15**(6): 660-666.

Fodor, J. A. (1975). The language of thought, Harvard Univ Pr.

French, R. M. (2003). "Catastrophic forgetting in connectionist networks." Encyclopedia of Cognitive Science.

Gallese, V. (2003). "The roots of empathy: the shared manifold hypothesis and the neural basis of intersubjectivity." Psychopathology **36**(4): 171-180.

Gavrila, D. M. (1999). "The visual analysis of human movement: A survey." Computer vision and image understanding **73**(1): 82-98.

Gentner, D. (2006). "Why verbs are hard to learn." Action meets word: How children learn verbs: 544-564.

Gerkey, B., R. T. Vaughan, et al. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems, Citeseer.

Ghazanfar, A. A., J. X. Maier, et al. (2005). "Multisensory integration of dynamic faces and voices in rhesus monkey auditory cortex." The Journal of neuroscience **25**(20): 5004-5012.

Gleitman, L. (1990). "The structural sources of verb meanings." Language acquisition **1**(1): 3-55.

Gollwitzer, P. M. (1993). "Goal achievement: The role of intentions." European review of social psychology **4**(1): 141-185.

Grafton, S. T., M. A. Arbib, et al. (1996). "Localization of grasp representations in humans by positron emission tomography." Experimental Brain Research **112**(1): 103-111.

Grand, S. and D. Cliff (1998). "Creatures: Entertainment software agents with artificial life." Autonomous Agents and Multi-Agent Systems **1**(1): 39-57.

Graziano, M. S. A. (1999). "Where is my arm? The relative role of vision and proprioception in the neuronal representation of limb position." Proceedings of the National Academy of Sciences **96**(18): 10418.

Grezes, J. and J. Decety (2001). "Functional anatomy of execution, mental simulation, observation, and verb generation of actions: a meta-analysis." Human brain mapping **12**(1): 1-19.

Grosz, B. and S. Kraus (1993). Collaborative plans for group activities, Citeseer.

Grosz, B. J. (1988). Plans for discourse, BBN LABS INC CAMBRIDGE MA.

Guizzo, E. (2011). "Robots with their heads in the clouds." Spectrum, IEEE **48**(3): 16-18.

Guye, M., G. J. M. Parker, et al. (2003). "Combined functional MRI and tractography to demonstrate the connectivity of the human primary motor cortex in vivo." Neuroimage **19**(4): 1349-1360.

Hagoort, P. (2005). "On Broca, brain, and binding: a new framework." Trends in Cognitive Sciences **9**(9): 416-423.

Hagoort, P., G. Baggio, et al. (2009). "Semantic unification." The cognitive neurosciences: 819-836.

Hagoort, P. and J. Van Berkum (2007). "Beyond the sentence given." Philosophical Transactions of the Royal Society B: Biological Sciences **362**(1481): 801-811.

Hamann, K., F. Warneken, et al. (2011). "Children's Developing Commitments to Joint Goals." Child Development.

Harnad, S. (1990). "The symbol grounding problem." Physica D: Nonlinear Phenomena **42**(1-3): 335-346.

Hay, D. F. (1979). "Cooperative interactions and sharing between very young children and their parents." Developmental Psychology **15**(6): 647.

Heylighen, F., M. Heath, et al. (2004). The Emergence of Distributed Cognition: a conceptual framework, Citeseer.

Hickok, G. (2009). "Eight problems for the mirror neuron theory of action understanding in monkeys and humans." Journal of cognitive neuroscience **21**(7): 1229-1243.

Hintzman, D. L. (1984). "MINERVA 2: A simulation model of human memory." Behavior Research Methods **16**(2): 96-101.

Hommel, B., J. Müsseler, et al. (2001). "The theory of event coding (TEC): A framework for perception and action planning." Behavioral and Brain Sciences **24**(05): 849-878.

Hopkins, C. D. (1974). "Electric communication in the reproductive behavior of Sternopygus macrurus (Gymnotoidei)." Zeitschrift für Tierpsychologie **35**(5): 518-535.

Howe, M. and R. Miikkulainen (2000). "Hebbian learning and temporary storage in the convergence-zone model of episodic memory." Neurocomputing **32**: 817-821.

Humayun, M. S., J. D. Weiland, et al. (2003). "Visual perception in a blind subject with a chronic microelectronic retinal prosthesis." Vision research **43**(24): 2573-2581.

Jacobs, R. A. (1988). "Increased rates of convergence through learning rate adaptation." Neural Networks **1**(4): 295-307.

Jacobs, R. A., M. I. Jordan, et al. (1991). "Adaptive mixtures of local experts." Neural Computation **3**(1): 79-87.

Jeannerod, M., M. A. Arbib, et al. (1995). "Grasping objects: the cortical mechanisms of visuomotor transformation." Trends in neurosciences **18**(7): 314-320.

Johnson, M. and Y. Demiris (2005). Hierarchies of coupled inverse and forward models for abstraction in robot action planning, recognition and imitation. AISB.

Jordan, M. I. and R. A. Jacobs (1994). "Hierarchical mixtures of experts and the EM algorithm." Neural Computation **6**(2): 181-214.

Kamewari, K., M. Kato, et al. (2005). "Six-and-a-half-month-old children positively attribute goals to human action and to humanoid-robot motion." Cognitive Development **20**(2): 303-320.

Kaski, S., J. Kangas, et al. (1998). "Bibliography of self-organizing map (SOM) papers: 1981-1997." Neural Computing Surveys **1**(3&4): 1-176.

Kawato, M. (1999). "Internal models for motor control and trajectory planning." Current opinion in neurobiology **9**(6): 718-727.

Kellman, P. J., E. S. Spelke, et al. (1986). "Infant perception of object unity from translatory motion in depth and vertical translation." Child Development: 72-86.

Kersten, D., P. Mamassian, et al. (1997). "Moving cast shadows induce apparent motion in depth." PERCEPTION-LONDON- **26**: 171-192.

Király, I., B. Jovanovic, et al. (2003). "The early origins of goal attribution in infancy." Consciousness and Cognition **12**(4): 752-769.

Knoblauch, A., H. Markert, et al. (2005). "An associative cortical model of language understanding and action planning." Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach: 72-111.

Koestler, A. (1968). "The ghost in the machine."

Kohler, E., C. Keysers, et al. (2002). "Hearing sounds, understanding actions: action representation in mirror neurons." Science **297**(5582): 846.

Kohonen, T. (1990). "The self-organizing map." Proceedings of the IEEE **78**(9): 1464-1480.

Kosslyn, S. M., W. L. Thompson, et al. (1995). "Topographical representations of mental images in primary visual cortex." Nature **378**(6556): 496-498.

Kotovsky, L. and R. Baillargeon (1998). "The development of calibration-based reasoning about collision events in young infants." Cognition **67**(3): 311-351.

Kramer, J. and M. Scheutz (2007). "Development environments for autonomous mobile robots: A survey." Autonomous Robots **22**(2): 101-132.

Lallée, S., J. Diard, et al. (2009). "Multiple Object Manipulation: is structural modularity necessary? A study of the MOSAIC and CARMA models."

Lallée, S., S. Lemaignan, et al. (2010). Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception. IROS, Taipei.

Lallée, S., S. Lemaignan, et al. (2011). Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions. IROS.

Lallée, S., C. Madden, et al. (2010). "Linking language with embodied teleological representations of action for humanoid cognition." Frontiers in Neurobotics.

Lallee, S., C. Madden, et al. (2010). "Linking language with embodied and teleological representations of action for humanoid cognition." Front Neurorobot **4**: 8.

Lallée, S., G. Metta, et al. (2009). Proprioception of the hand contributes to visual recognition speed and accuracy: Evidence from the Multi-Modal Convergence Map model of Parietal Cortex Area 5. SFN, Chicago.

Lallée, S., F. Warneken, et al. (2009). Learning to collaborate by observation. Epirob, Venice.

Lallee, S., E. Yoshida, et al. (2010). "Human-robot cooperation based on interaction learning." From Motor Learning to Interaction Learning in Robots: 491-536.

Langley, P., J. E. Laird, et al. (2009). "Cognitive architectures: Research issues and challenges." Cognitive Systems Research **10**(2): 141-160.

Laureys, S., M. Boly, et al. (2006). "Tracking the recovery of consciousness from coma." Journal of Clinical Investigation **116**(7): 1823.

Lefort, M., Y. Boniface, et al. (2010). "Multi-sensory integration by constrained self-organization."

Lefort, M., Y. Boniface, et al. (2010). "Self-organization of neural maps using a modulated BCM rule within a multimodal architecture."

Lefort, M., Y. Boniface, et al. (2011). "Self-organizing neural maps for multi-modal associations." BMC Neuroscience **12**(Suppl 1): P125.

Lemaignan, S., R. Ros, et al. (2010). ORO, a knowledge management platform for cognitive architectures in robotics, IEEE.

Leonardo, M., J. Fieldman, et al. (1995). "A functional magnetic resonance imaging study of cortical regions associated with motor task execution and motor ideation in humans." Human brain mapping **3**(2): 83-92.

Lewin, K. (1951). "Intention, will and need."

Liberman, A. M. and I. G. Mattingly (1985). "The motor theory of speech perception revised." Cognition **21**(1): 1-36.

Lingnau, A., B. Gesierich, et al. (2009). "Asymmetric fMRI adaptation reveals no evidence for mirror neurons in humans." Proceedings of the National Academy of Sciences **106**(24): 9925.

Lipton, P. A., P. Alvarez, et al. (1999). "Crossmodal associative memory representations in rodent orbitofrontal cortex." Neuron **22**(2): 349-359.

Llinás, R. and U. Ribary (1994). "Perception as an oneiric-like state modulated by the senses." Large-scale neuronal theories of the brain: 111-124.

Lotze, M., P. Montoya, et al. (1999). "Activation of cortical and cerebellar motor areas during executed and imagined hand movements: an fMRI study." Journal of Cognitive Neuroscience **11**(5): 491-501.

Magnin, M., M. Rey, et al. (2010). "Thalamic deactivation at sleep onset precedes that of the cerebral cortex in humans." Proceedings of the National Academy of Sciences **107**(8): 3829.

Magoulas, G. D., M. N. Vrahatis, et al. (1999). "Improving the convergence of the backpropagation algorithm using learning rate adaptation methods." Neural Computation **11**(7): 1769-1796.

Maguire, M. J., K. Hirsh-Pasek, et al. (2006). "14. A Unified Theory of Word Learning: Putting Verb Acquisition in Context." Action meets word: How children learn verbs: 364.

Mamassian, P., D. C. Knill, et al. (1998). "The perception of cast shadows." Trends in Cognitive Sciences **2**(8): 288-295.

Mandler, J. (1992). "How to build a baby: II. Conceptual primitives." PSYCHOLOGICAL REVIEW-NEW YORK- **99**: 587-587.

Mandler, J. M. (1992). "How to build a baby: II. Conceptual primitives." PSYCHOLOGICAL REVIEW-NEW YORK- **99**: 587-587.

Maravita, A., C. Spence, et al. (2003). "Multisensory integration and the body schema: close to hand and within reach." Current Biology **13**(13): R531-R539.

Markov, N., M. M. Ercsey-Ravasz, et al. (2010). "Principles of inter-areal connections of the macaque cortex."

Markov, N., P. Misery, et al. (2011). "Weight consistency specifies regularities of macaque cortical networks." Cerebral Cortex **21**(6): 1254.

Mataric, M. J., M. Williamson, et al. (1998). Behavior-based primitives for articulated control. Fifth international conference on simulation of adaptive behavior on From animals to animats 5.

Maturana, H. R. and F. J. Varela (1987). The tree of knowledge: The biological roots of human understanding, New Science Library/Shambhala Publications.

McGurk, H. and J. MacDonald (1976). "Hearing lips and seeing voices."

Meltzoff, A. N. and M. K. Moore (1977). "Imitation of facial and manual gestures by human neonates." Science **198**(4312): 75.

Meltzoff, A. N. and M. K. Moore (1983). "Newborn infants imitate adult facial gestures." Child Development **54**(3): 702-709.

Meltzoff, A. N. and M. K. Moore (1989). "Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms." Developmental Psychology **25**(6): 954.

Ménard, O. and H. Frezza-Buet (2005). "Model of multi-modal cortical processing: Coherent learning in self-organizing modules." Neural Networks **18**(5-6): 646-655.

Meredith, M. A. and B. E. Stein (1986). "Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration." Journal of Neurophysiology **56**(3): 640-662.

Metman, L. V., J. S. Bellevich, et al. (1993). "Topographic mapping of human motor cortex with transcranial magnetic stimulation: Homunculus revisited." Brain topography **6**(1): 13-19.

Metta, G., G. Sandini, et al. (2006). "Understanding mirror neurons: a bio-robotic approach." Interaction studies **7**(2): 197-232.

Mintz, T. H. (2003). "Frequent frames as a cue for grammatical categories in child directed speech." Cognition **90**(1): 91-117.

Moeslund, T. B. and E. Granum (2001). "A survey of computer vision-based human motion capture." Computer vision and image understanding **81**(3): 231-268.

Moll, M. and R. Miikkulainen (1997). "Convergence-zone episodic memory: Analysis and simulations." Neural Networks **10**(6): 1017-1036.

Moll, M., R. Miikkulainen, et al. (1994). The capacity of convergence-zone episodic memory, IEEE.

Morrow, J. D. and P. Khosla (2002). Manipulation task primitives for composing robot skills. ICRA, Albuquerque, IEEE.

Mussa-Ivaldi, F. and E. Bizzi (2000). "Motor learning through the combination of primitives." Philosophical Transactions of the Royal Society B: Biological Sciences **355**(1404): 1755.

Mussa-Ivaldi, F. A., S. F. Giszter, et al. (1994). "Linear combinations of primitives in vertebrate motor control." Proceedings of the National Academy of Sciences of the United States of America **91**(16): 7534.

Nehaniv, C. L. and K. Dautenhahn (2002). "2 The Correspondence Problem." Imitation in animals and artifacts: 41.

Noy, N. F. and M. A. Musen (2004). "Ontology versioning in an ontology management framework." Intelligent Systems, IEEE **19**(4): 6-13.

Nunn, J. A., L. J. Gregory, et al. (2002). "Functional magnetic resonance imaging of synesthesia: activation of V 4/V 8 by spoken words." Nature neuroscience **5**(4): 371-375.

Oshii, M., M. Shirow, et al. (2004). Ghost in the Shell, Manga Entertainment.

Oudeyer, P. Y., F. Kaplan, et al. (2007). "Intrinsic motivation systems for autonomous mental development." Evolutionary Computation, IEEE Transactions on **11**(2): 265-286.

Paine, R. W. and J. Tani (2004). "Motor primitive and sequence self-organization in a hierarchical recurrent neural network." Neural Networks **17**(8-9): 1291-1309.

Papliński, A. and L. Gustafsson (2005). "Multimodal feedforward self-organizing maps." Computational Intelligence and Security: 81-88.

Papliński, A. and L. Gustafsson (2006). "Feedback in multimodal self-organizing networks enhances perception of corrupted stimuli." AI 2006: Advances in Artificial Intelligence: 19-28.

Pei, S. C. and Y. S. Lo (1998). "Color image compression and limited display using self-organization Kohonen map." Circuits and Systems for Video Technology, IEEE Transactions on **8**(2): 191-205.

Pellegrino, G., L. Fadiga, et al. (1992). "Understanding motor events: a neurophysiological study." Experimental brain research **91**(1): 176-180.

Perani, D., S. Cappa, et al. (1995). "Different neural systems for the recognition of animals and man-made tools." NeuroReport **6**(12): 1637.

Plagianakos, V., G. Magoulas, et al. (2001). "Learning rate adaptation in stochastic gradient descent." NONCONVEX OPTIMIZATION AND ITS APPLICATIONS **54**: 433-444.

Porro, C. A., M. P. Francescato, et al. (1996). "Primary motor and sensory cortex activation during motor performance and motor imagery: a functional magnetic resonance imaging study." The Journal of neuroscience **16**(23): 7688.

Pulvermüller, F. (1999). "Words in the brain's language." Behavioral and brain sciences **22**: 253-279.

Pulvermüller, F. (2005). "Brain mechanisms linking language and action." Nature Reviews Neuroscience **6**(7): 576-582.

Quigley, M., B. Gerkey, et al. (2009). ROS: an open-source Robot Operating System.

Raichle, M. E., A. M. MacLeod, et al. (2001). "A default mode of brain function." Proceedings of the National Academy of Sciences **98**(2): 676.

Rich, C., C. L. Sidner, et al. (2001). "Collagen: applying collaborative discourse theory to human-computer interaction." AI magazine **22**(4): 15-26.

Rizzolatti, G. and M. A. Arbib (1998). "Language within our grasp." Trends in neurosciences **21**(5): 188-194.

Rizzolatti, G. and L. Craighero (2004). "The mirror-neuron system." Annu. Rev. Neurosci. **27**: 169-192.

Röder, B., F. Rösler, et al. (2004). "Early vision impairs tactile perception in the blind." Current Biology **14**(2): 121-124.

Ross, H. S. (1982). "Establishment of social games among toddlers." Developmental Psychology **18**(4): 509.

Roth, M., J. Decety, et al. (1996). "Possible involvement of primary motor cortex in mentally simulated movement: a functional magnetic resonance imaging study." NeuroReport **7**(7): 1280.

Ruby, P. and J. Decety (2001). "Effect of subjective perspective taking during simulation of action: a PET investigation of agency." Nature neuroscience **4**(5): 546.

Rui, Y. and P. Anandan (2000). Segmenting visual actions based on spatio-temporal motion patterns, IEEE.

Saylor, M. M., D. A. Baldwin, et al. (2007). "Infants' on-line segmentation of dynamic human action." Journal of Cognition and Development **8**(1): 113-128.

Scassellati, B. (1999). "Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot." Computation for metaphors, analogy, and agents: 176-195.

Schaal, S., A. Ijspeert, et al. (2003). "Computational approaches to motor learning by imitation." Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences **358**(1431): 537.

Schall, J. D., A. Morel, et al. (1995). "Topography of visual cortex connections with frontal eye field in macaque: convergence and segregation of processing streams." The Journal of neuroscience **15**(6): 4464.

Schuldt, C., I. Laptev, et al. (2004). Recognizing human actions: A local SVM approach, IEEE.

Searle, J. R. (1980). "Minds, brains, and programs." Behavioral and Brain Sciences **3**(03): 417-424.

Searle, J. R. (1990). "Collective Intentions and Actions John R. Searle." Intentions in communication: 401.

Sentis, L. and O. Khatib (2005). "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives." International Journal of Humanoid Robotics **2**(4): 505-518.

Shi, Q., L. Wang, et al. (2008). Discriminative human action segmentation and recognition using semi-markov model, IEEE.

Shirow, M., M. Oshii, et al. (1995). Ghost in the Shell, Dark Horse Comics.

Shmuelof, L. and E. Zohary (2005). "Dissociation between ventral and dorsal fMRI activation during object and action recognition." Neuron **47**(3): 457-470.

Siskind, J. (1998). Visual event perception.

Siskind, J. (2001). "Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic." Journal of Artificial Intelligence Research **15**(1): 31-90.

Siskind, J. M. (1998). Visual event perception. NEC Research Symposium.

Sommer, F. T. and T. Wennekers (2003). "Models of distributed associative memory networks in the brain." Theory in Biosciences **122**(1): 55-69.

Spelke, E. S. (1990). "Principles of object perception." Cognitive science **14**(1): 29-56.

Spelke, E. S., P. Vishton, et al. (1995). "10 Object Perception, Object-directed Action, and Physical Knowledge in Infancy."

Sporns, O. and G. M. Edelman (1993). "Solving Bernstein's problem: A proposal for the development of coordinated movement by selection." Child Development **64**(4): 960-981.

Sporns, O., G. Tononi, et al. (2000). "Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices." Cerebral Cortex **10**(2): 127.

Steels, L. and J. Baillie (2003). "Shared grounding of event descriptions by autonomous robots." Robotics and Autonomous Systems **43**(2-3): 163-173.

Steels, L., F. Kaplan, et al. (2002). "Crucial factors in the origins of word-meaning." The transition to language **12**: 252-271.

Taga, M. E. and B. L. Bassler (2003). "Chemical communication among bacteria." Proceedings of the National Academy of Sciences of the United States of America **100**(Suppl 2): 14549.

Talmy, L. (1988). "Force dynamics in language and cognition." Cognitive science **12**(1): 49-100.

Tenorth, M., A. Perzylo, et al. "The RoboEarth language: Representing and Exchanging Knowledge about Actions, Objects, and Environments."

Tettamanti, M., G. Buccino, et al. (2005). "Listening to action-related sentences activates fronto-parietal motor circuits." Journal of cognitive neuroscience **17**(2): 273-281.

Thelen, E. (1995). "Motor development: A new synthesis." American Psychologist **50**(2): 79.

Thomas, U., B. Finkemeyer, et al. (2003). Error-tolerant execution of complex robot tasks based on skill primitives. ICRA, Taipei, IEEE.

Thorpe, S., R. Guyonneau, et al. (2004). "SpikeNet: Real-time visual processing with one spike per neuron." Neurocomputing **58**: 857-864.

Tomasello, M. (1995). "Joint attention as social cognition." Joint attention: Its origins and role in development: 103–130.

Tomasello, M. (1999). The cultural origins of human cognition, Harvard University Press Cambridge, MA.

Tomasello, M. (2009). Why we cooperate, The MIT Press.

Tomasello, M., M. Carpenter, et al. (2005). "Understanding and sharing intentions: The origins of cultural cognition." Behavioral and Brain Sciences **28**(05): 675-691.

Tomasello, M. and M. J. Farrar (1986). "Joint attention and early language." Child development: 1454-1463.

Trevarthen, C. (1979). "Communication and cooperation in early infancy: A description of primary intersubjectivity." Before speech: The beginning of interpersonal communication: 321–347.

Tsakiris, M. and P. Haggard (2005). "The rubber hand illusion revisited: visuotactile integration and self-attribution." Journal of Experimental Psychology: Human Perception and Performance **31**(1): 80.

Tuomela, R. (2001). Collective intentionality and social agents.

Tuomela, R. (2005). "We-intentions revisited." Philosophical Studies **125**(3): 327-369.

Velleman, J. D. (1997). "How to share an intention." Philosophy and Phenomenological Research **57**(1): 29-50.

Vernon, D., G. Metta, et al. (2007). The icub cognitive architecture: Interactive development in a humanoid robot, Ieee.

Vernon, D., G. Metta, et al. (2007). "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents." Evolutionary Computation, IEEE Transactions on **11**(2): 151-180.

Vezoli, J., M. A. Gariel, et al. (2010). "Hierarchical order in the cortex."

Völkel, M. and T. Groza (2006). SemVersion: RDF-based ontology versioning system.

Waibel, M., M. Beetz, et al. (2011). "RoboEarth." Robotics & Automation Magazine, IEEE **18**(2): 69-82.

Warneken, F., F. Chen, et al. (2006). "Cooperative activities in young children and chimpanzees." Child Development **77**(3): 640-663.

Warneken, F. and M. Tomasello (2007). "Helping and cooperation at 14 months of age." Infancy **11**(3): 271-294.

Weick, K. E. and K. H. Roberts (1993). "Collective mind in organizations: Heedful interrelating on flight decks." Administrative science quarterly: 357-381.

Willems, R. M., A. Özyürek, et al. (2007). "When language meets action: The neural integration of gesture and speech." Cerebral Cortex **17**(10): 2322-2333.

Williamson, M. M. (1996). Postural primitives: Interactive behavior for a humanoid robot arm. Fourth international conference on simulation of adaptive behavior on From animals to animats 4.

Wilson, M. (2002). "Six views of embodied cognition." Psychonomic Bulletin & Review **9**(4): 625-636.

Woodward, A. L. (1999). "Infants' ability to distinguish between purposeful and non-purposeful behaviors." Infant Behavior and Development **22**(2): 145-160.

Wyatt, T. D. (2003). Pheromones and animal behaviour, Cambridge University Press Cambridge.

Wyatt, T. D. (2003). Pheromones and animal behaviour: communication by smell and taste, Cambridge Univ Pr.

Zwaan, R. A. and C. J. Madden (2005). "Embodied sentence comprehension." Grounding cognition: The role of perception and action in memory, language, and thinking: 224-245.

Zweigle, O., R. van de Molengraft, et al. (2009). RoboEarth: connecting robots worldwide, ACM.