DELIVERABLE D3
SYSTEMS ENGINEERING SPECIFICATION
OCTOBER 2008

CHRIS

Cooperative Human Robot Interaction Systems

Produced by WP3:
Peter Ford Dominey, Zhenli Lu, Stephane Lallee

Grant Agreement number: 215805

Project acronym: CHRIS

Project title: Cooperative Human Robot Interaction Systems

Project start date: 01 March 2008

Funding Scheme: Small or medium-scale focused research project (STREP)

Date of latest version of Annex I against which the assessment will be made: 14th Dec 2007

Project coordinator name, title and organization:
Prof Chris Melhuish, Director, Bristol Robotic Laboratory (BRL), University of the West of England, Bristol

Tel: ++44 (0) 117 32 86334

Fax: ++44 (0) 117 32 82688

E-mail: Chris.Melhuish@brl.ac.uk

Project website address: http://www.chrisfp7.eu/

Table of Contents

Please note:

At the date of this document preparation, the Dominey research team are in the process of transferring the contract from Lyon2 to INSERM. The Lyon group, which has produced D3, has been based at INSERM from 10th October 2008 (which is the date the institutions have been set to agree effective start/end of contract).

We recognize and appreciate the contribution of Lyon2 to date, and the support they have provided to enable this. Please note that future work referred to in this document by the Dominey research team, is assumed to be performed at INSERM, on the assumption that the contract transfer is completed successfully.

Roadmap

This deliverable is intended to define the major functional components of the system, their interfaces and to assign responsibility to partners for their implementation.

- Chapter 1 is an updated/annotated version of the preliminary working document. It is included for historical information purposes, and to define the overview of the « canonical table building scenario ».

- Chapter 2 defines the current conception of the system. As agreed, functional components of the system are to be implemented in the YARP framework. High level communication interfaces can access these functions via YARP ports.

- Chapter 3 provides a walkthrough of the system during a particular part of an interaction.

- Appendix 1 contains a more detailed breakdown of the table building scenario.

- Appendix 2 contains a proposed (but not imposed) flow chart.

- Appendix 3 is a signature page, in which each institution technical representative (WP leader) will sign-off, indicating that they have read the document, they agree in principal, and if necessary that they have certain reserves which can be written in the provided space.

1. Preliminary System Analysis

1.1 Methodology

1. Define a "canonical" scenario, and perform a scripted walkthrough (the Situated Scenario Enactment) of that scenario in order to validate the global concept that we are to implement. This was achieved at Meeting 2 in Genoa.
2. Based on that walkthrough,
    - identify the functions that will implement this behavior, and allocate those functions to tasks within the technical work packages.
    - Define the interfaces between the functions

The current section 1 represents an initial pass of this activity.
3. Circulate this specification for comment, before delivery of the final version, D3 – Functional Requirements and Interface Agreement Document.
4. All partners sign off on the document.

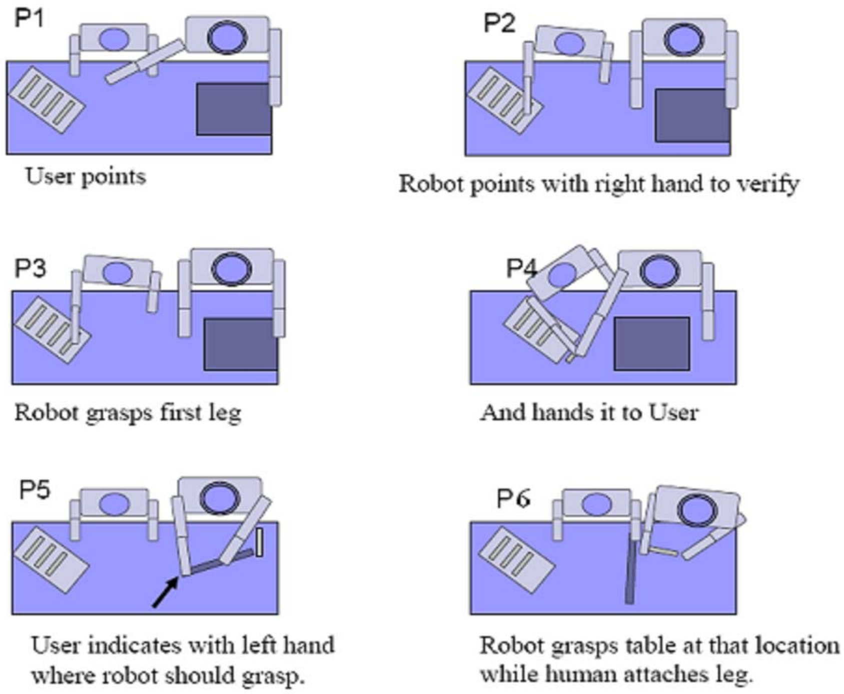

Figure 1. Schematic overview of initial portion of the interaction scenario.

Figure 1 is a graphic reminder of the outline of the scenario that we have analyzed together. Following Figure 1 is a breakdown of the interaction, and an initial allocation of the functions to different WP tasks.

Overview of the Scenario from the MPG perspective of Cooperation: This canonical scenario entails some of the crucial behavioral elements required for cooperative interaction as defined in the conceptual framework provided by the MPI. The current scenario is an example for a dyadic collaborative act in an expert-novice interaction with agents performing asymmetric complementary roles (see CHRIS vocabulary on CHRIS WIKI). It entails critical verbal and non-verbal communicative acts, such as referential gestures (pointing) used in a imperative-communicative function (see P1 & P2). Moreover, it entails an act of requested sharing in P3 & P4). Perhaps most importantly, the scenario ends with a joint coordinated activity in which the two agents have to perform two interdependent and complementary roles synchronously to succeed (P5 & P6).

1.2 Disclaimers

1. This document addresses implementation issues. In this context, we note that WP6 deals with the fundamentals of human-human interaction which will be modeled and essential components implemented on the robot platform. As such, with complex cognitive reasoning underpinning much of the robot, and all of the human behavior, tables 1 and 2 assume that results of WP6 are implemented on the platform, and as such only capability deriving from robot design and build WPs has been explicitly listed.

2. This analysis focuses on the "integrated platform" demonstration with the iCub, so specifics related to Jido, HRP2 and BERT robots are not directly addressed.

3. At the simplest level, for visually guided grasping, the robot will be pre-trained on the grasp for a set of objects. The objects will be placed in “optimal” grasping configurations in the scenario. Object learning will consist simply in attaching names to these objects. More complex methods including object categorization can also be addressed.

4. The specification proposed in this document represents our current vision of the system, and this will mature over time. It is a first iteration, focused on the needs on a set up based on the iCub. It will be used as an operational example. The architecture as well as the possible interactions (nature, associated data structures, temporal constraints) between the main components will be incrementally refined together with a more detailed description of the internals of each component. One aspect that we consider as essential would be the incremental definition of a "data base" that represents all the information effectively manipulated and exchanged by the various software components. The more mature specification will be provided in D5 System Engineering Analysis report - Month 17 (Jul 09).

5. The purpose of D3 is to provide a high level specification of the system architecture and functional requirements based on the SSE. However, it is beyond the scope of this document to specify in detail the implementation of different functions at the WP level. Those details will be provided by the WPs and will be included in D5.

Table 1. Breakdown of interaction. Note that in this table, column 3 refers to task specifications from the detailed WP descriptions. H: Human, R: Robot. Items marked in Bold were not actually present in the Situated Simulation Enactment..

Scenario Description	Function	Allocation to WP Tasks.
H: Enters room and sits next to robot	**Robot recognizes human**	**T4.3 Visual analysis of behavior** **T7.3 Body/hand Facial gesture**
R: Oh, Hello Chris.	**Robot orients to user, and physically and verbally acknowledges his presence**	**T7.3 Body/hand Facial gesture**
H: <u>Give*</u> me one of those legs	Spoken language recognition . Can be provided by the RAD toolkit.	T5.1 Engagement management (includes spoken language interface)
R: Sorry, I don’t know what a leg looks like *Access to object database has no entry for lexical item leg*	Handling uncertainty: The Get(X) behavior fails because X is unknown. This activates a contingency in the Get(X) plan (SHARY)	T5.2 Uncertainty management
H: <shows the robot by pointing> This is a leg	Visual following of the hand, and then identification	T4.3 visual analysis of behavior

	of the closest recognizable object	T7.3 Body gesture recognition
R: <Points to a the indicated leg> Is that it? *Requires visually guided pointing.*	Robot points and uses head orientation to indicate the object. This must be done safely, taking into account the human (avoiding collision, etc.)	T4.1-4.3 visually guided motion T7.2 Safe interaction
H: Yes Robot now associates word "leg" with this visually recognized object.	Vocabulary acquisition.	T5.1 Engagement management (vocabulary for spoken language).
R: OK, here you go <picks up the leg	Reach to grasp (P3) Right arm grasp	T4.1-3 Visually guided action
> <and passes it to the human> Requires knowledge of what it means to "give"	Pass right (P4)	T5.1-3 "Give" behavior in SHARY
Robot recognizes that human is too far away, and that is must transfer the object from one hand to another in order to pass it	**This requires decisional planning in order to deal with changes in the physical state of execution.**	**T5.1-3 "Give" behavior in SHARY**
H: Now can you <u>hold</u> this table here <indicates with his own hand where the robot should grasp the table>	Robot visually identifies the user's hand's configuration,	T4.3 Mirroring and imitation
R: <grasps the table> Like this?	and uses this in order to guide its own hand to that location to then hold the table.	T4.3 Mirroring and imitation
H: Yes that's good. <attaches the leg>	confirmation	T5.1 Engagement management
H: Ok you can let go now <robot releases the table>		T5.1 Engagement management

Anticipation and learning: As the successive legs are attached, the robot uses the interaction history to compare ongoing behavior with previous behavior in order to predict and anticipate.

1.3 Variations on the "canonical" scenario

1. Greeting the user (inserted above)

2. Need to modify a behavior (passing the leg when the user is out of reach) provided by decisional planning (inserted above).
3. Need to modify a learned behavior: User observes that the robot is grasping the leg with a faulty grasp, and so initiates a dialog to edit this behavior.
4. Nonverbal communication: Holding the hand extended, palm up, is recognized as a "give me" command.
5. Nonverbal communication: Holding the hand extended, palm pointing towards the robot, is recognized as a "halt/pause" command.
6. Robot recognizes that it is having a failure (e.g. the torso motor fails to turn) and asks the human to intervene.
7. Robot recognizes that it is having a failure and asks human if it should continue in graceful degradation mode.
8. Compliant motion: Human and robot hold the table together, and human compliantly guides the robot so that together they place the table in a particular final configuration.

1.4 Preliminary allocation of functions to the hardware and software systems

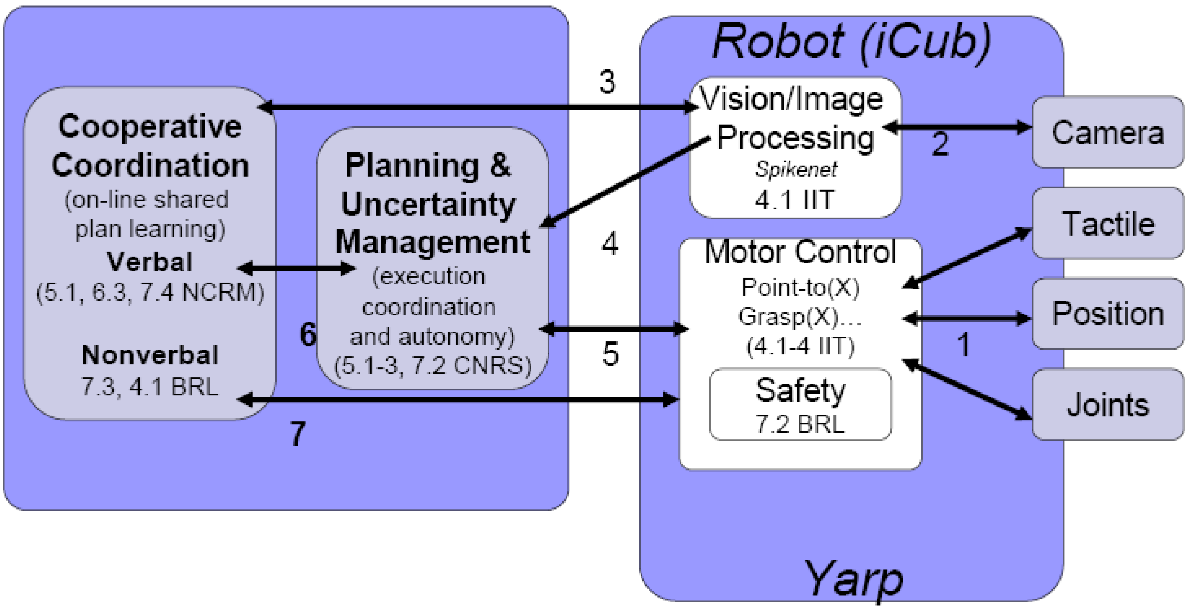

Figure 2. Functional Connectivity

Figure 2 provides a preliminary allocation of functions, defining functional modules and their corresponding WP tasks, and also identifying a responsible partner for each major function. In the text below, the modules are described, and the principal interfaces.

Motor Control (IIT, BRL):
Implements basic visually guided behaviors, and low level safety, including: Localize(X), Grasp(X), Point-to(X).

Vision/Image processing (IIT):

Implements recognition and localization. One candidate is the Spikenet system that we are currently investigating. Different labs (CNRS,BRL, IIT, NCRM) might use different local systems.

Planning & Uncertainty Management (CNRS):
Implements higher level decisional planning capabilities, which yields a pre-defined set of robust behaviors: A set of rich behavioral building blocks will be predefined and implemented to form a repertoire of behaviors. These behaviors will include …. Get (X), Hold(X); Put(X at Y), Localize (X), etc. These are behaviors that rely on perception and heavily rely on decisional planning. That is, they will take into account the user's engagement, the possible obstacles or states that must be considered in order to achieve the objective. So, specifically, Get(X) will rely on the behavior Grasp(X), but it will also rely on decisional planning which may require the robot to uncover the object, ask the user where the object is, move to a different location, etc. This corresponds to the SHARY system of CNRS.

Cooperative Coordination (NCRM, BRL)
The user will have access to these behaviors via spoken language. Through interaction with the user, the robot can acquire an "interaction history", i.e. a recording of previous interactions that can be used to allow the robot to predict and more efficiently participate in current and future interactions. This will result in the learning of new plans. These plans will be made up of actions at the "robust behavior" level. That is, a plan may include "Get(screwdriver)" but this action may be accomplished in different ways, depending on the physical state, and the ability to deal with uncertainty.

1.5 Module Interfaces

1. Physical devices to controllers. These interfaces are internal to the different robot platforms.
2. Camera to image processing. Example: Spikenet vision processing reads video at 30Hz frame-rate.
3a, 4. Vision processing to Cooperative Coordination & Planning and Uncertainty management: X,Y coordinates (and/or 3D robot centered coordinates) of recognized objects. This includes manipulable objects as well as the different hand configurations (pointing, palm up to grasp, palm forward to halt, etc.)
3b. Cooperative Coordination to Vision processing: specification of recognition library (allows selective, context dependant vision processing).
5. Planning & Uncertainty Management – Motor Control: Based on the plan execution state, P&UM will invoke motor control functions. Return status will allow monitoring of plan execution status.
6. Cooperative Coordination – Planning & Uncertainty Management: High level commands (e.g. Give me the ball) which will rely on autonomous management of the human-robot interaction.

7. Cooperative Coordination – Motor Control: Lower level commands, such as to take a posture, orient the head etc. can be issued directly.

2. UPDATED SYSTEM ARCHITECTURE

Based on the preliminary system analysis above, we have now developed a more detailed system definition. In particular the functional components have been defined in terms of YARP ports and their related commands.

2.1 System Architecture Synthesis

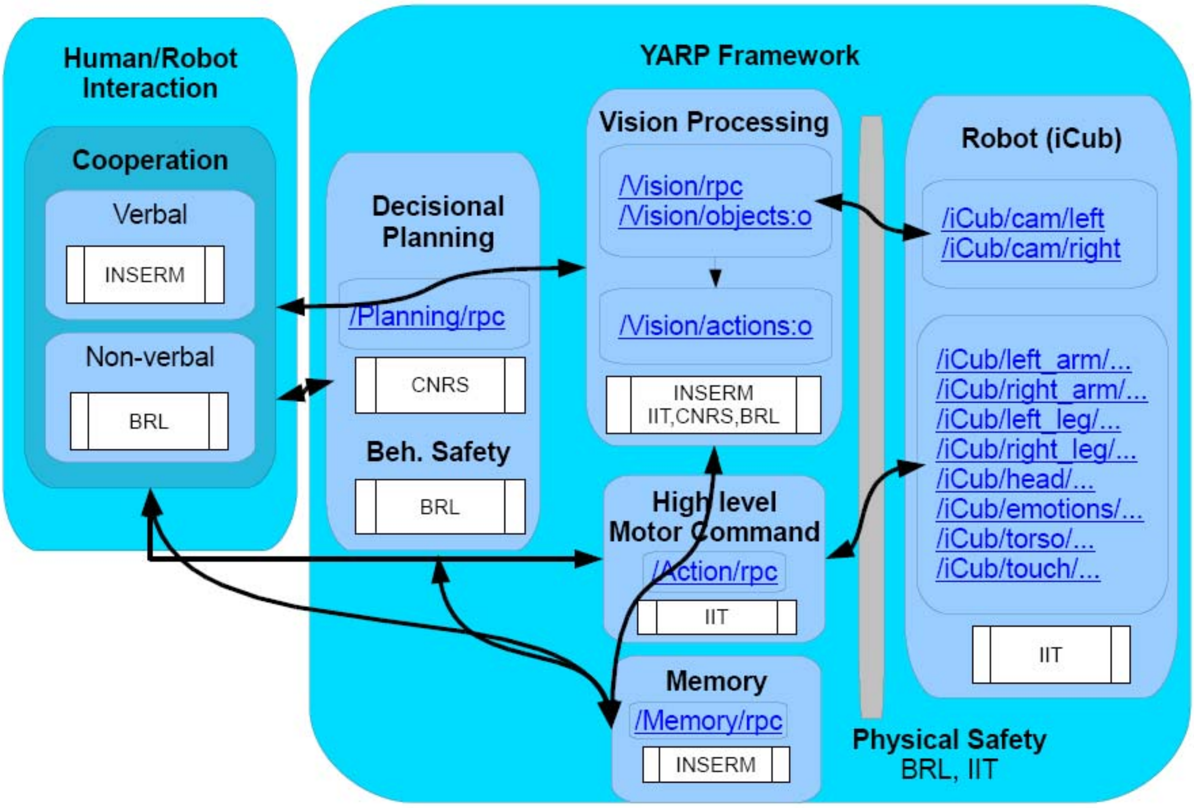

Figure 3 : System architecture synthesis. Note that this figure reflects, as specified in Disclaimer 2, an analysis based on the "integrated platform" demonstration with the iCub. This should generalize across the project, and apply to the largest extent possible to the different platforms including BERT, Jido and HRP2.

2.2 YARP Framework detail

YARP will be the communication protocol between modules. It provides a mechanism by which functions can be accessed by different "users" via well defined interfaces. Figure 3 defines the minimum set of YARP ports used to connect the modules. Every module team will define his own « private ports » but the high level communication should be done by reading/writing these ports. The Vision Chapter gives a more detailed definition of a module, including private ports.

Each port called /module_name/rpc is a way to send high level text commands to a module. Here is the minimum set of commands that should be provided by each of these ports.

The following sections will specify each of the indicated functional components, with its interfaces, and responsibility for implementation.

High Level Motor Command (IIT – iCub, CNRS – Jido, BRL - BERT)

The robot will have a set of implemented motor commands, that can be initiated via calls to the YARP port /Action/rpc. Each robot site will be responsible for this for their robot. As specific commands become implemented, this list will be expanded. It shall also specify pre- and post-conditions, inputs and outputs. Here we provide an illustrative example of actions.

/Action/rpc

- grasp hand(left, right) object_name
  - e.g. “grasp left cup” - grasp the cup with the left hand.
  - reply OK or an error code (too far, too heavy...)

- release hand(left, right)
  - open the hand to release an object
  - reply OK or an error code

- point hand(left, right) object_name
  - reply OK or an error code

- reach hand(left, right) object_name
  - reply OK or an error code

- orient body_part(gaze, head, trunk) object_name
  - reply Ok or an error code

- go_to location
  - reply Ok or an error code

Shared Memory (INSERM, IIT, BRL, CNRS):

Shared memory corresponds to state-related information that is available within the system. This will include the names of known objects, of known behaviors, an interaction history or list of action that have been performed or observed by the system.

This shared memory (or Database) will be the information structure backbone of the system. For Deliverable 5 we will provide a more complete specification of the contents of this. We have created a

Each of the technical workpackages which creates and or uses information INSERM will be responsible for creating the structure for this capability and different functional modules will be responsible for updating it appropriately (see paragraph 2.3).

/Memory/rpc

- get object object_name
  - reply
    - detail about the object, if the object is known
    - error code if the object is not known
    - a list of all the known objects if object_name is "all"

- get behavior behaviour_name
  - reply
    - sequence of actions if the behavior is known
    - error code if the behavior is not known
    - a list of all the known behaviors if behaviour_name is "all"

- get action_history number (what is this?)
  - reply the last *number* events occurred

- set object object_name object_details
  - add/modify an object
  - reply OK or an error code

- set behavior behaviour_name action_list
  - add/modify a behavior
  - reply OK or an error code

Decisional Planning (CNRS): (see internal Deliverable D5.1 Adaptive Planning Capability Report)

Decisional Planning: A crucial aspect of the human-robot interaction within the CHRIS project will be that the robot takes into account the presence of the human, and the context of the physical environment. Concretely this means that the system may be required to respond to changes in the physical environment, or the human's status, with replanning. Thus in contexts where there is the possibility of uncertainty, the "grasp left cup" command must be "validated" by the decisional planning capability, to ensure that the conditions for its execution are satisfied. In case they are not satisfied, decisional planning will take over to ensure that they can be (as defined in Alami et. al. 2007, Sisbot et al 2008). This capability is provided by CNRS. Likewise, decisional planning will implement predefined cooperation behaviors.

Decisional planning will be implemented as a Decisional control system that will have to select and parameterize robot behaviors and actions. The main aspect here is to produce a collaborative robot that takes into account the presence of the human, and the context of the physical environment. Concretely this means that the system may be required to respond to changes in the physical environment, or the human's status.

The framework is a continuous planning process that achieves context dependent task refinement. Plans will be produced that allow the robot to participate to a joint activity with a person and to maintain common ground through a set of multi-modal communication acts that support the interactive task achievement. Proper behavior includes pertinent initiative taking as well as the ability to exhibit its intentionality and to comply with human needs and preferences. These capabilities will be based on work conducted in cooperation with MPG in WP6.

This component will essentially access a data base maintained by the different robot modules and select parameterize robot sensory-motor behaviors and communication actions.

Safety (BRL, CNRS, IIT): (see internal deliverable D7.1 Requirements specification for a physically and behaviorally safe service robot).

A second crucial aspect of these interactions is their safety. Control of the force exerted, human-robot proximity, introduction of fail-safe mechanisms in the case of problems and related measures must be taken into consideration. These capabilities are undertaken by BRL, with contribution from CNRS and IIT.

The safety aspects of the robotic system will be implemented at different system levels depending on the required reaction speed. A crucial distinction is made between physical and behavioural safety.

Model based monitoring of the internal states of the robot's actuators and sensors, continuous evaluation of the integrity of the real-time communication links as well as fast post collision reactions all form part of the physical safety system. These mechanisms will be implemented at the lowest level possible and have reflex-like (real-time) qualities which do not require links with higher level systems to function properly. However, actions and analysis of the physical safety system will be propagated to the higher level planning and behavioural safety module.

The behavioural safety modules are mainly concerned with the prevention of harmful collisions during interaction. These high level functionalities can be looked at as a filter between the motion planner and the high level motor command generator. The behavioural safety module receives information from the motion planner, the external environment, the human operator (e.g. position, gaze, attention) and the robot's internal health status (as provided by the physical safety system). Based on this information, "collision risk indices" will be calculated and possible deviations from the planer's original trajectory may be introduced by the system. These deviations may take the form of spatial or temporal corrections of the original trajectory, including complete trajectory cancelation if desirable.

/Planning/rpc

- execute behaviour_name
  - reply OK or an error code (Can't do that without hurting a human...)
- get possibility behaviour_name
  - reply OK or an error code (same as execute but without doing the action)

Vision Processing

To interact with the human, the robot must be able to see: It should be able to identify a set of known objects. It should be able to identify a set of known human actions and postures. At different sites, vision may be implemented differently (e.g. INSERM and IIT will likely use

Spikenet, BRL may use a combination of video images and marker detection, etc.). Note that for action recognition, the visual component is described in section 3.2. However, in some cases the use may specify via language the action he/she is performing, so the system should accommodate a multimodal action recognition capability.

/Vision/rpc

- get position object_name
  - reply
    - (x,y,z) of each instance of object_name
    - error code if the object is not visible
    - list of all the recognized objects and their positions if object_name is "all"

- get orientation object_name
  - reply
    - orientation of each instance of object_name
    - error code if the object is not visible
    - list of all the recognized objects and their orientations if object_name is "all"

Cooperation

This function provides the high level verbal and non-verbal interface to the system. Spoken language can be used to execute commands (which can be verified by Decisional Planning) and to initiate cooperative behavior acquisition, as illustrated in the Scripted Scenario Enactment. The resulting cooperative plans will be added to the shared memory, for subsequent future access. This capability is initially implemented in the CSLU Rad system for use on the iCub. This does not preclude the use of other speech processing software.

Nov-verbal cooperation will include the observation of specific human gestures such as the palm vertical « stop » command (BRL).

Note that Cooperation is distributed: The Decisional Planning function implements cooperation, as it will monitor human engagement, and respond accordingly (CNRS).

The behavioral and communicative skills to successfully cooperate in this fashion are also seen as building blocks for successful cooperation in other types of situations. Namely, a further future goal of the project could be to add the feature of "role-reversal" to the system, so that the two agents can freely switch between the currently fixed roles of expert and novice, enabling forms of symmetric collaboration among two equally skilled agents.

2.3 Memory content specification

Shared memory corresponds to state-related information that is available within the system. This will include the names of known objects, of known behaviors, an interaction history or list of action that have been performed or observed by the system.

Object : physical objects known by the robot. An object is at least composed of :

names *an object may have multiple names*

ID *the "system name" used by the recognition process* grasping configuration *need to define what it can be*

There will initially be a set of known objects including
Hand, leg, etc.

Behaviors : Sequences of actions known by the robot. In a simple way, behaviors can be lists of actions similar to those used in Dominey, Mallet, Yoshida (2007).
There will be an initial set of behaviors composed of only one action to allow the user to command the robot. Theses primary behaviors will be equivalent the actions provided by /Action/rpc.

Note that for all of these behaviors, the execution of the action will be managed by "Planning and Uncertainty Management" to determine if the current context allows the behavior to be executed, or if some replanning must be performed.

Interaction History:

Sequential history of all interactions. This can be queried and used for statistical or other kinds of learning for behavior extraction (as in the paper by Dominey and Metta at Humanoids 2008). The memorization process of this history should be automatic and transparent.

3. DETAILED INTERACTION ANALYSIS

3.1 More Detailed Breakdown, step 9-14 (Pseudo Code)

Annex 1 provides a detailed breakdown of the table building scenario. In this section, we examine in close detail the steps 9-14 which involve a series of human-robot interactions which provide a fairly complete exercise of the system. The analysis is in a form of pseudo-code which refers to the different functions. Recall that both annex 1 and this analysis are not to be considered binding, but rather as illustrative examples.

9 : H: Give* me a leg#.

Speech_Recognition → give(leg, me)
query /Planning/rpc
execute give(leg, me)
query /Memory/rpc
get behavior give → {grasp left argument, reach left recipient, release left}
get object leg → Error : Not known
get object me → {"me" (x,y,z) (x,y,z)}

The object leg is not a part of the robot memory
2reply Error : Object unknown "leg"
Enter the cooperation sub-routine "Learn an object"

10 : R: Can you show* me a leg#?

Speech_Synthesis("Show me a leg") → wait

11 : H: This is a leg* right here.

Speech_Recognition → stop waiting

Reading the /Vision/actions:o broadcast until the right action is recognized
read /Vision/actions:o → point_to(human, unknown_object_ID)

Ask confirmation by showing the object
query /Actions/rpc
1point_to (object_ID)

12/13 : R : Is it what you mean ?
H : Yes.

Speech_Synthesis("Is it what you mean ?")
Speech_Recognition("yes") → yes

Add knowledge of a leg to the memory
query /Memory/rpc
1set object "leg" object_ID
Exit the subroutine "Learn an object"

14 : R : Ok, I will give it to you.

execute (give(leg, me))
query /Memory/rpc
get behavior give → {grasp left argument, reach left recipient, release left}
get object leg → {"leg (x,y,z) (x,y,z) }
get object me → {"me" (x,y,z) (x,y,z)}

Everything is ok, execute the actions of the behavior
query /Action/rpc
grasp(right, leg) → Ok
reach(right, me) → Error : Too far

Execution of reach failed. Replanning find a solution by a passing from one hand to another
Replanning (give(leg, me), "Too far")
query /Action/rpc
reach(right, left_hand) → Ok
open(right) → Ok
grasp(left, Object_ID) → Ok
reach(left, me) → Ok
open(left) → Ok
orient(body, neutral) → Ok
Speech_Synthesis "What is next ?"

3.2 The vision processing module

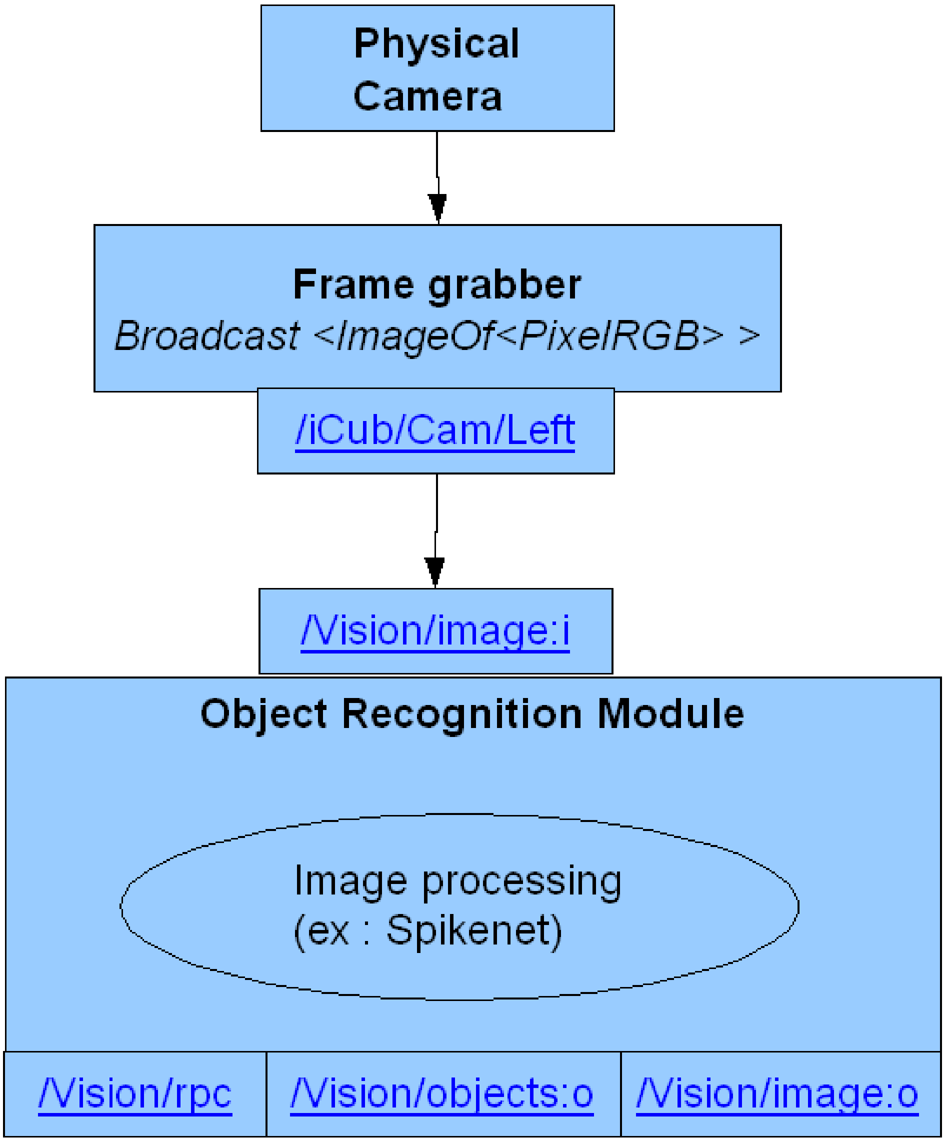

Figure 4 : Camera/Object recognition interface

Vision processing is a crucial component of human-robot cooperation. In this section we describe a proposal for the implementation of vision processing. The image recognition processing will be performed using commercial software Spikenet. A license distribution agreement has been offered by Spikenet for 1500 euros per/site for use of software with the spikenet.dll compiled. Concretely, to use the compiled software requires a hardware dongle.

Generic interface for the object recognition process.
Composed of 4 YARP ports :

- /Vision/image:i is used to get the images from the frame grabber module (typically it will be connected to /iCub/Cam/)

- /Vision/rpc is the port which will read and treat text queries. Minimal set of available queries should be :
  - get position object_name (reply (x,y,z) of each instance of object_name)
  - get list (reply with the list of recognized objects)

- /Vision/objects:o is a port which is always broadcasting the name and positions of all the recognized objects in the form *time name X Y Z*
  Example :
  10:30:10 Banana 0 1 1 Orange 0 3 2
  10:30:11 Banana 0 1 1 Orange 0 3 1
  10:30:12 Banana 0 1 1 Orange 0 3 0
  ...
  Such a broadcasting allow the detection of motion and the time stamp can provide information for speed estimation.

- /Vision/image:o broadcasts images with stuff drawn on it (ex : markers or labels on recognized objects) for visualization purposes.

INSERM will provide a template of the code in C++ to create this interface for use with Spikenet software on the iCub.

Visual Action Recognition Module :

The robot needs to be able to recognize not only objects but also actions proceeded by a human or another robot.
Such a module can be useful in a lot of way :

- non verbal interaction (halt, point...)
- learning by observation ("watch me while I do this task")
- mirror activation (mirror neuron modeling)

As the main input for this module can be the objects positions, we propose it as an extension of the Object Recognition Module but it's possible to make it a part of a more global Vision Module. Note that action identification can also come from the user saying what she is doing – i.e. a form of multimodal action recognition.

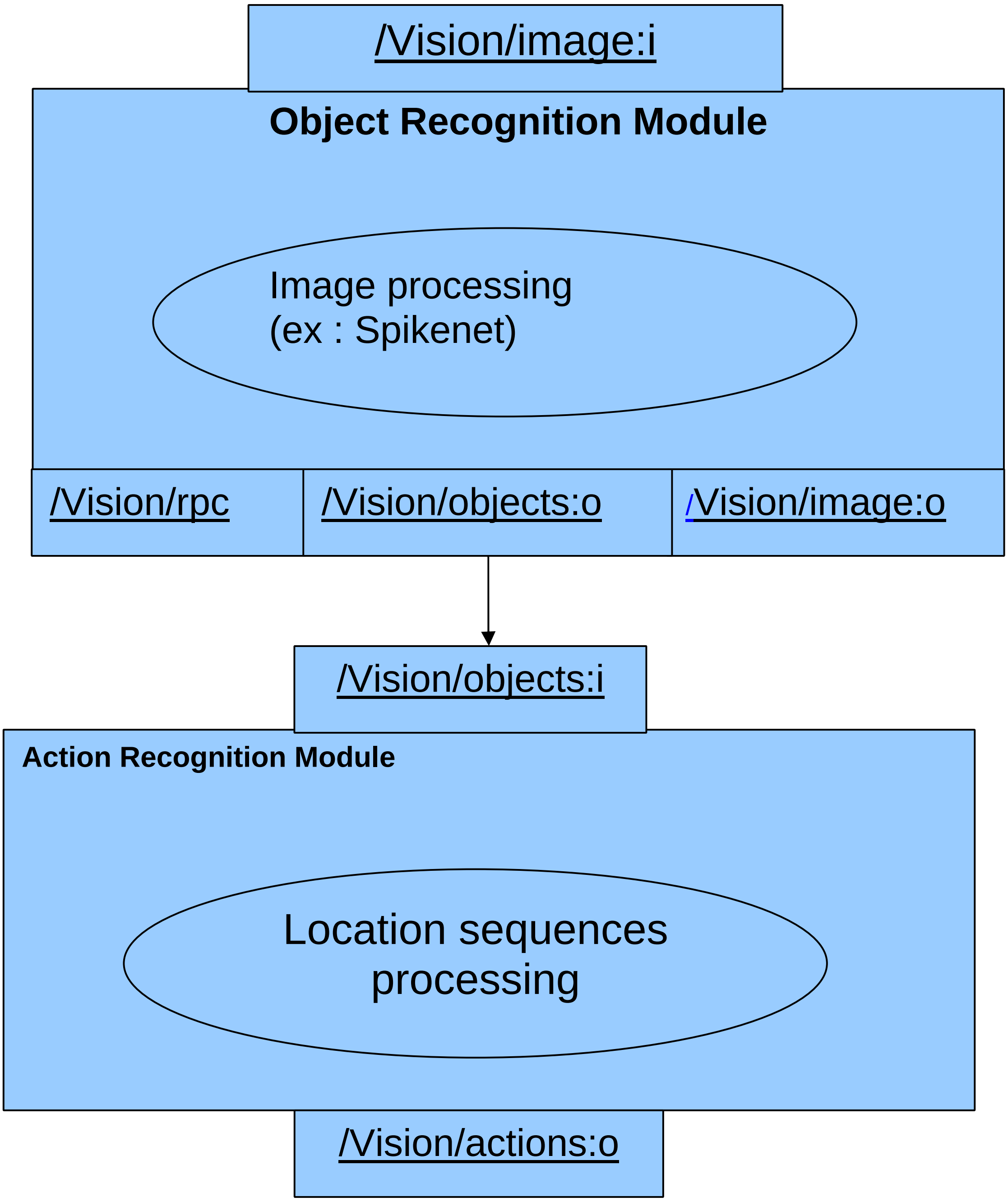

Figure 5 : Object/Action recognition interface

/Vision/Actions:o should be broadcasting the recognized actions marked with a time stamp. More high level processes (shared plan, observation learning...) will listen to this port and analyze the action flow. The broadcasting should be of the form *time action arguments_list*

Example :

10:30:00 grasp robot leg
10:31:00 reach robot left_arm human_hand
10:32:00 grasp human leg

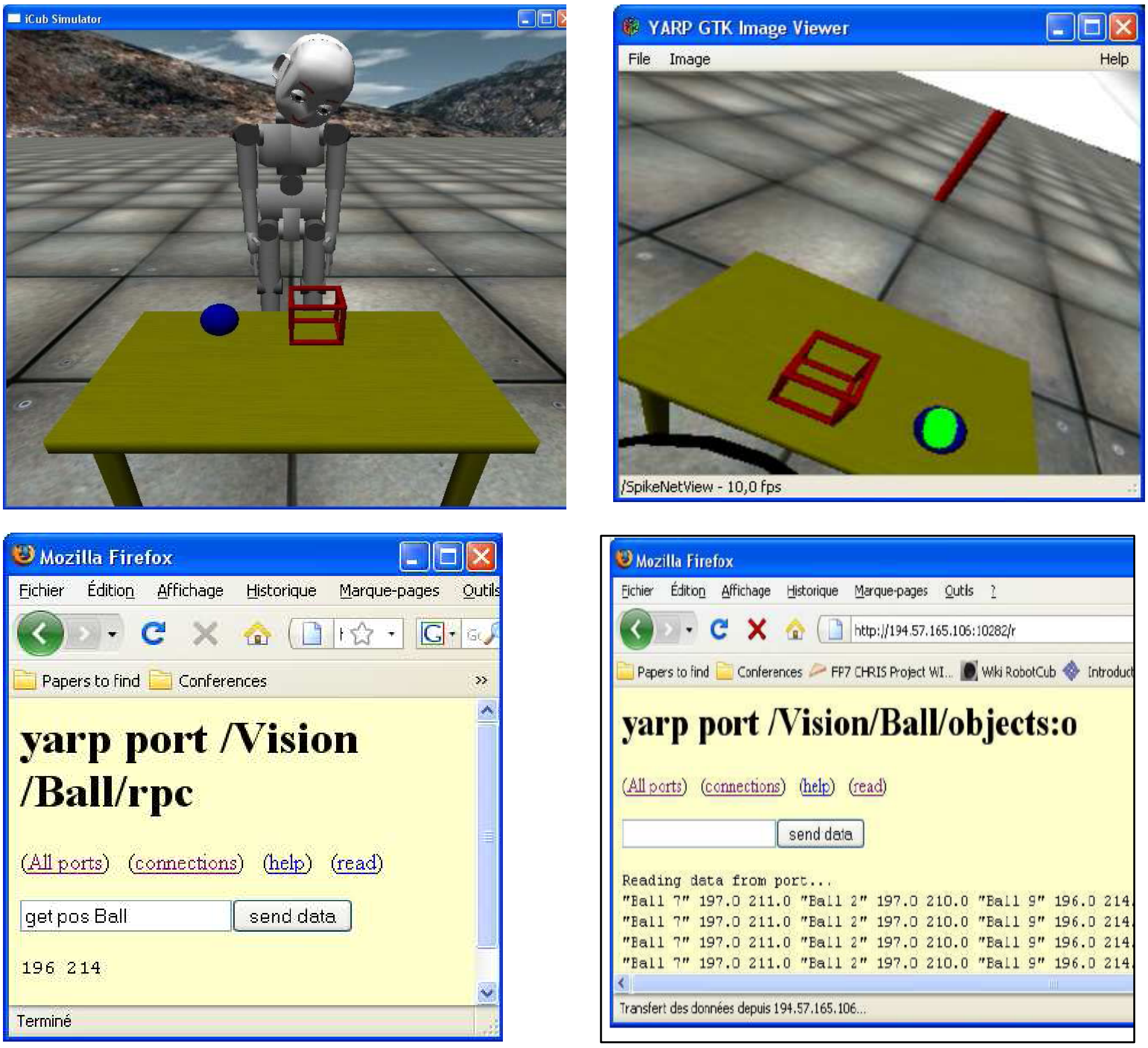

Figure 6. Prototype vision system implemented on iCub simulator. Upper left: robot configuration. Upper right: robot view of table surface, with Spikenet recognition results (green circle) superimposed on recognized object. Lower left: rpc port query and response for recognized ball. Lower right: continuous recognition via :o port.

Annex 1 – Detailed Scenario Breakdown

This breakdown of interaction of the "canonical" scenario represents a possible interaction framework. It should not be considered as limiting other possibilities. Again, WP6 deals with the fundamentals of human-human interaction which will be modeled and essential components implemented on the robot platform. As such, with complex cognitive reasoning underpinning much of the robot, and all of the human behavior, this table assumes that results of WP6 are implemented on the platform, and as such only capability deriving from robot design and build WPs has been explicitly listed

R : Robot, H: Human (Chris (XXX)), SD: Scenario description, FS: Function Specification, TS: Task Specification.

<>: action, Italic: Comment, *: Action, #: Object, underline: keywords, boldface: optional function.

Table 2. Breakdown of interaction in the "canonical" scenario for CHRIS project.

No	SD	FS	TS
1	Start	System initialization.	T5.1 Engagement management.
2	**R: <idle state >**	**Idle gesture.**	**T5.1 Engagement management.**
3	**H: <Enters room and sits next to robot>**	**Robot recognizes human.**	**T4.3 Visual analysis of behavior.** **T7.3 Body/ hand Facial gesture.**
4	**R: Hello who are you?**	**Robot orients to the user, and physically and verbally acknowledge his presence.**	**T5.2 Uncertainty management.** **T7.3 body/hand Facial gesture.**
5	**H: I am Chris.**	**Spoken language recognition. Can be provided by the RAD toolkit.** **Visual recognized the face.**	**T4.3 Visual analysis of face.** **T7.3 Face recognition.**
6	R: What can I do for you, (wish what)Chris?	Handling uncertainty. *waiting for the task command input.*	T5.2 Uncertainty management.
7	H: Yes, Build* this table#.	Spoken language recognize.	T5.1 Engagement management(including spoken language interface).
8	R: what should I do now?	Handling uncertainty: The task build(X) behavior fails because X is unknown. This	T5.2 Uncertainty management.

		activates a contingency in the Build(X) plan (SHARY)	
9	H: Give* me a leg#.	Spoken language recognize.	T5.1 Engagement management(including spoken language interface). T5.3 Goal and Decision Planning.
10	R: Can you show* me a leg#?	Handling uncertainty: The Get(X) behavior fails because X is unknown. This activates a contingency in the Get(X) plan (SHARY)	T5.1 Engagement management. T5.2 Uncertainty management. T7.3 Body/hand Facial gesture.
11	H: This is a leg* right here. <shows the robot by pointing*>.	Visual following of the hand and then identification of the closest recognizable object.	T4.3 Visual analysis of the behavior. T5.1 Engagement management. T7.3 Body gesture recognition.
12	R: Is it what you mean? < reach* to the leg , then point* to the leg>	Points to the indicated leg. Requires visually guided pointing.	T4.3 Mirroring behaviors. T4.4Intention understanding and sequencing behaviors. T5.1Engagement management. T5.2 Uncertainty management. T7.2 Safe interaction. T7.3 Body/hand facial gesture. T7.4 Limited verbal goal-negotiation.
13	H: That is it. <nodding*>	Vocabulary acquisition. Visual analysis of the body language.	T4.3 Visual analysis of the behavior. T5.1 Engagement management. T7.3 Body gesture recognition.
14	R: OK, I will take* it. < Pick* up the leg.> *Reach* and grasp* the leg#. if needed the leg will be passed to the other hand.*	This action requires decisional planning in order to deal with changes in the physical state of execution. write down this step into the history database.	T4.3 Visually guided action. T5.1Engagement management. T7.2 Safe interaction. T7.3 Body/hand facial gesture.
15	R: What is next?	Teaching requirements	T5.2 Uncertainty management.
16	H: Pass* it to me.	Spoken language recognize. action analysis.	T5.1 Engagement management(including spoken language interface). T5.3 Goal and Decision Planning.
17	R: Passing* the leg.	H-R co-operation. add this	T5.1 Engagement management.

	<Robot pass* the leg to human>	action into the sequence.	T5.3 Goal and Decision Planning. T7.2 Safe interaction.
18	R: what is next? (Now what?)	Teaching requirements.	T5.2 Uncertainty management.
19	H: Hold* this table right here. <indicates with his own hand where the robot should hold the table>	Robot visually identifies the user's hand's configuration.	T4.3 Mirroring and imitation. T4.4 Intention understanding and sequencing behaviors. T7.3 Body gesture recognition .
20	R: <Hold* the table> H: <Install the leg>	H-R co-operation, Mirroring and imitation, complaisant cooperative, write down this action to the database.	T5.1 Engagement management.
21	R: Like* this?	To check the gesture is right or not.	T5.2 Uncertainty management.
22	H: Yes, That is great.	Confirmation, write down this to the database and knowledge sequence.	T5.1 Engagement management.
23	R: What should I do now?	Teaching requirements	T5.2 Uncertainty management.
24	H: ok, You can release* the table*.	Release is a reverse action of hold.	T4.3 Mirroring behaviors. T4.4Intention understanding and sequencing behaviors. T5.1 Engagement management.
25	R: Releasing* the table. <robot return to is start gesture> H: <Place the table on the desk.>	The robot return to its start gesture.	T5.1 Engagement management. T7.2 Safe interaction.
26	R: what can I do now?	Teaching inquiry	T5.2 Uncertainty management.
27	H: Pass* me another leg#.	Index from the knowledge sequence, check out its a repeat action the automatically execute the knowledge sequence. find another leg on the desk.	T4.3 Visual analysis of behavior. T5.1 Engagement management.
28	R: Approaching* leg# two.	Implementation	T5.1 Engagement management. T5.3 Goal and Decision Planning. T7.2 safe interaction.
29	R: Grasping *leg two.	Implementation	T5.1 Engagement management.

			T5.3 Goal and Decision Planning. T7.2 Safe interaction.
30	R: So I give* it to you?	When co-operation the robot needs a inquiry.	T5.2 Uncertainty management .
31	H: Yes.	Confirmation, execute the next action in the knowledge sequence.	T5.1 Engagement management.
32	R: Passing* the leg. H-R<pass* and receive* leg two. >	Implementation.	T5.1 Engagement management.
33	R: So I hold* the table?	Co-operation inquiry.	T5.2 Uncertainty management .
34	H: Yes, good idea. <H-R: Hold* the table and fix leg two>.	Confirmation, execute the next action in the knowledge sequence.	T5.1 Engagement management.
35	R: Should I release* the table now?	Co-operation inquiry.	T5.2 Uncertainty management.
36	H: No, once again. <repeat Hold* the table and fix leg two >	Disavowal, repeat command.	T5.1 Engagement management.
37	R: Now?	Co-operation inquiry.	T5.2 Uncertainty management.
38	H: ok .	Confirmation.	T5.1 Engagement management.
39	R:<robot return to is start gesture.>	Execute the next action in the knowledge sequence.	T5.1 Engagement management. T7.2 Safe interaction.
40	R: I think I understand*.	Robot find another leg on the desk then inspired the robot to try the learn and repeat task. Send an inquiry.	T5.2 Uncertainty management.
41	H: ok .	Confirmation.	T5.1 Engagement management.
42	R: Let me try*?	Send an inquiry.	T5.2 Uncertainty management.
43	H: Go ahead*.	Confirmation.	T5.1 Engagement management.
44	R:Grasping* like me? <robot grasp the third leg>	Send an inquiry.	T5.2 Uncertainty. Management.
45	H: good.	Confirmation.	T5.1 Engagement management.
46	R: passing* the leg. <H-R:pass and receive the leg>	Send an inquiry. then execute the action.	T5.1 Engagement management.
47	R: Holding* the table.	Send an inquiry. then execute the action.	T5.1 Engagement management.
48	R: Releasing* the table	Send an inquiry. then execute the action.	T5.1 Engagement management.
49	H: ok. <Place the table on the desk>	Confirmation (one small task finished).	T5.1 Engagement management.

	R: <return to the start posture>		
50	R: How did I do?	Send an inquiry.	T5.2 Uncertainty management.
51	H: That is great.	Confirmation.	T5.1 Engagement management.
52	R: taking* the last leg.	Find there is still one leg on the desk then automatically repeat the acting sequence.	T5.1 Engagement management.
53	R: Getting* the last leg .	Send an inquiry. then execute the action.	T5.1 Engagement management.
54	R: Passing* the leg . <pass and receive the leg>	Send an inquiry. then execute the action.	T5.1 Engagement management.
55	R:holding* the table.	Repeat.	T5.1 Engagement management.
56	R: Releasing* the table.	Repeat.	T5.1 Engagement management.
57	R: <return* to its start gesture>.	Repeat.	T5.1 Engagement management.
58	R: what is the next? <find* no leg on the desk>	Send an inquiry.	T5.2 Uncertainty management.
59	H: Stop*. This is a table. <shows the robot by pointing*>.	Visual following of the hand and then identification of the closest recognizable object.	T4.3 visual analysis of the behavior. T5.1 Engagement management. T7.3 Body gesture recognition.
60	R:Well,I now what a table# is. That is the goal# we got.	End task.	T5.1 Engagement management. T7.3 Body/hand Facial gesture.
61	H: good. Let us put the table on the desk.	Confirmation.	T5.1 Engagement management.
62	**H: Hold the table like me, Let us put the table on the desk.**	**New task.**	**T4.4Intention understanding and sequencing behaviors.** **T5.1 Engagement management.**
63	**R:OK.<H-R: hold and place the table on the desk.>**	**Intention understanding and complaisance motion control.**	**T4.4Intention understanding and sequencing behaviors.** **T5.1 Engagement management.**
64	H: Thank you. <H-R:shake hands>	End task.	T5.1 Engagement management.
65	R: You are welcome. <Robot return to its idle gesture>	Waiting for new task.	T5.1 Engagement management.

Annex 2 – Interaction Flow Chart

This flow chart represents a possible interaction framework. It should not be considered as limiting other possibility.

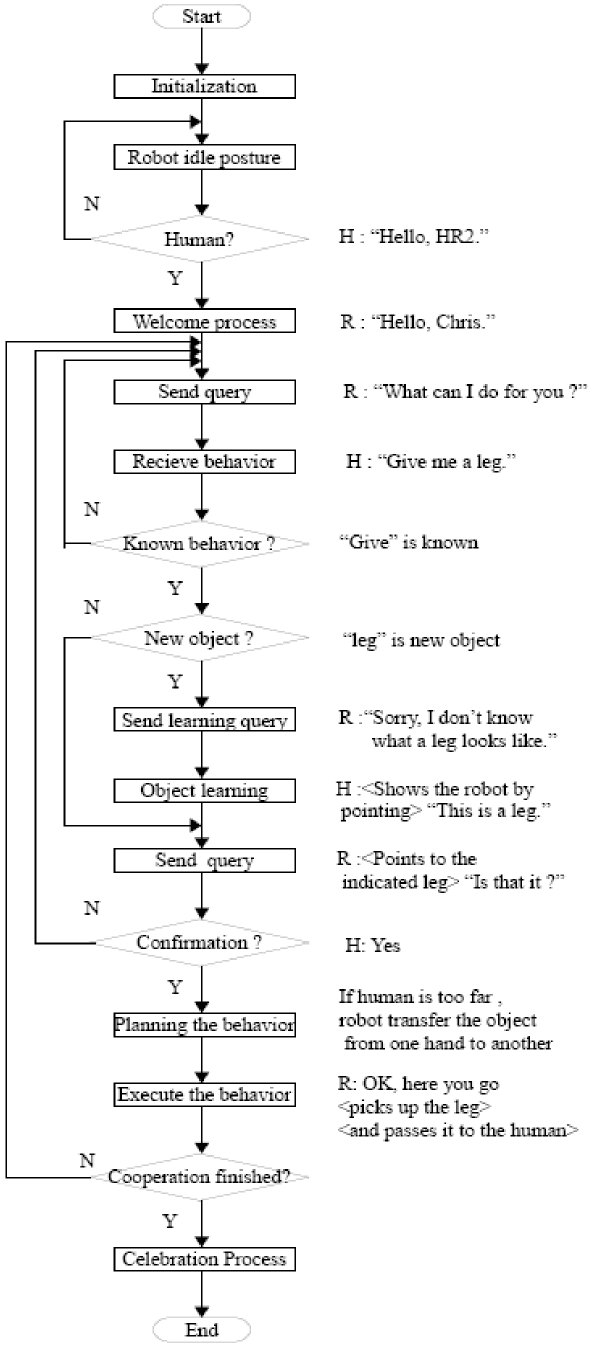

Figure 6. System flowchart of the robot.

Figure 6 is the system flowchart of the robot ,and the specification is shown as following:

Step 1. When the robot is powered on, the initialization process is executed so as to initialize the hardware device and software parameters, and then generate a system function report.

Step 2. Based on the report, the robot can display the status messages and switch itself to idle posture.
Step 3.When perceived a human by vision and spoken language recognition<Human sit down nearby the chair> H: "Hello, HR2."
Step 4.Robot runs a welcome process. For example, R: "Hello, Chris."

Step 5. Robot sends out some co-operation inquiry. For example, R: "What can I do for you?"
Step 6. Robot waits for some behavior command from the human, for example, H: "Give me a leg."

Step 7.Robot checks the behavior is a known one or not. If the behavior is not mastery then go to step 5. (For example, "give" is a known behavior, then go to step 8.)

Step 8.Robot checks the object is a new one or not. If it is not a new object then go to step 11. (For example, "leg" is an unknown object, then go to step 9.)

Step 9.Robot sends an learning inquiry. (For example, R: Sorry, I don't know what a leg looks like? then go to step 10.)

Step 10.Object learn process, Robot receive the object information through visual and spoken language recognition, For example, H :<Shows the robot by pointing> "This is a leg."

Step 11.Accept and deal with the learning result; send an inquiry. For example, R :<Points to the indicated leg> "Is that it ?"
Step 12.Accept and deal with the inquiry result; if it is a confirmation then it executes behavior, else go to step.5. For example, H :<Points to the indicated leg> "Yes", then go to step 13.

Step 13.Planning the behavior. For example, R: recognize that human is too far away, and that it must transfer the object from one hand to another in order to pass it.

Step 14.Executing the behavior. For example, R : "OK, here you go"<pick up the leg>,<pass it to the human>.

Step 15.Memorize the new behavior and add all executed behaviors in behavior history.

Step 16. If the cooperation is not finished, go to Step 5; else execute the celebration process.

Step 17. End.

References

1. Sisbot, Clodic, Alami, Ransan (2008) Supervision and motion planning for a mobile manipulator interacting with humans, Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction
2. R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, R. Chatila "Toward Human-Aware Robot Task Planning", AAAI Spring Symposium, Stanford, USA, March 2006
3. A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila "SHARY: a supervision system adapted to Human-Robot Interaction", ISER 2008, Athens, July 2008.
4. Samir Alili, Rachid Alami, and Vincent Montreuil, "A Task Planner for an Autonomous Social Robot", DARS-2008, Japan, November 2008.
5. P-F. Dominey, A. Mallet, and E. Yoshida. Progress in programming the hrp-2 humanoid using spoken language. In IEEE International Conference on Robotics and Automation, Roma (Italy), April 2007. IEEE.
6. P-F. Dominey, G. Metta, F. Nori, L. Natale. Anticipation and Initiative in Human-Humanoid Interaction. IEEE Humanoids 2008