# ABSTRACT

The main aim of this project is to explore the safety and liveness requirements in human robot interaction. This project was carried out at the CHRIS Project area at Bristol Robotics Laboratory, and to some extent the two projects have the same objects of study, which are cooperative human robot interaction systems.

Various issues are investigated and many devices and techniques are studied in this project. For example, the main points of this project, which are safety properties and liveness properties in human robot interaction, are carefully researched. In order to verify these findings, some scenarios in the context of a drink servant robot are specified and implemented, thus these safety and liveness requirements can be concretized and tested.

Experiments are carried out not only to look for initial as well as boundary values of some specified actions, but also to test if the states of the system and the procedures of the scenarios comply with the safety and liveness requirements. After the experiments, many rules are found and validated, which can form the basis of a set of new laws in safe human-assist robots in the future.

Although most of the intended goals of this project have been achieved, the formal specifications of limits and constraints and formal verification were replaced by a validation of scenarios due to time limitation. Suggestions of future work are also given lastly.

# TABLE OF CONTENT

# Chapter 1: Introduction

## 1.1 Aims and Objectives

This project aims to explore methodologies or approaches applicable to robots which are designed to assist humans. Issues to be investigated cover both safety as well as liveness of the robot when in contact with humans, with the former meaning that robots operate within their spec and within safe limits while the latter means that robots do not fail to do the things they are supposed to do.

The objective of the project is to investigate these issues in the context of the robot being an adaptable learning system, in other words, to identify requirements that help establish the fact that the safety and liveness properties specified for this robot are not violated while the robot is acquiring new knowledge.

The subject for the study is a drink servant robot with a head that is capable of wide-field in-scene head location, head-direction and gaze-checking, although in this project it is used only for gaze-checking; and an associated freely adaptive robot arm with hand, to be specific, with the grip. Besides, a voice system will be used to communicate with the users, in the most intuitive way.

Original specific objectives relating to the above scenarios are:

1. To identify potentially dangerous behavioral adaptations of this robot that should be avoided.

2. To identify the safety reflexes that can be built into this robot to limit behavioral adaptations within safe bounds.

3. To explore the effects such limits have on constraining adaptive behavior, ensuring these limits still guarantee useful adaptations.

4. To validate these findings in a practical case study that:

- Implements a situation that addresses one safety reflex with the aim of preventing a particular 'dangerous' behavior;

- Explores different settings to identify the best compromise between making the system safe yet not fully restricting its adaptability;

- To formally specify the limits and constraints identified, for example, in the form of formal properties or assertions;

- To verify by simulation/test or through other formal means that the implementation satisfies the specified limits and constraints.

5. To evaluate the case study.

Most parts of these objectives have been achieved except the formal specification of the limits and constraints and the verification work, which has been replaced by a validation of the scenarios, for instance, test of the implementation and calibration, in the project.

## 1.2 Organization of the Dissertation

The dissertation mainly consists of five parts.

Firstly, Chapter 2 introduces background knowledge of this thesis as well as providing state of the art information. In addition to Aimov's three laws of robotics, which have been a staple of science fiction, it also gives an overview of hazards, and explains safety and liveness properties which are the two main points this dissertation focuses on. Some previous work which relates to this project is also mentioned.

Chapter 3 is the specification of this project, and introduces what the project intends to do but not exact what has been done. And it also demonstrates the context and the object of this project as well as the concept of adaptable behaviors, or in other words, the concept of behavioral adaptations.

Chapter 4 is the detailed introduction of the project's design and implementation. It presents an overview of the whole system using methods such as a state diagram, and then providing detailed descriptions of every scenario, every module, every device and component.

The experimental phase is next given, and Chapter 5 mainly introduces the aims, the settings as well as the processes of the experiments, the way to give rise rules, and finally many rules and values are listed as the results are reached.

Lastly, a conclusion of the project is given and some ideas about future work are provided.

# Chapter 2: Background Knowledge and State-of-the-Art

## 2.1 Introduction

This chapter is devoted to introduce the background and related knowledge of this study, which focuses mainly on the concepts of safety, the concept of liveness as well as the safety principles in human-robot interaction and in the design of human assist robotics. Besides, some related and prior work is also showed in this chapter, including the CHIRIS Project which is still under construction at Bristol Robotics Laboratory at the moment.

## 2.2 Background

As the object of the study is human assist robot, so firstly, we would like to introduce the background of robotics, especially those designed to assist humans, as well as the current co-existence status of the human and the robot.

According to the common understanding, "Robot is an automatic device that performs functions normally ascribed to humans or a machine in the form of a human". The word has actually a Czech root which use for forced labor or serf. The Czech playwright, Karl Capek, use it first time in his play in 1921 [1].

The concept of using robots came to humans' mind from the moment that human thought they can put some time consuming and repetitive activities on the shoulder of some other creatures/devices that can do the job restlessly but correctly.

Robots as physical assistance to humans must lessen stress and fatigue, increase human capabilities in terms of force, speed, and precision, and try to improve the quality of life generally; also, human can bring experience, global knowledge, and understanding for a correct execution of tasks [2].

Invention of robots goes back to 250 to 200 BC [3] which for the first time a water clock with movable arm had been designed. Ctesibius of Alexandria builds organs and water clocks with movable figures.

Gradually the idea formed with the intelligent thoughts to make a feasible and functional device to help the human with some understanding of the activity that the device does itself and work in harmony with people assisting them with different tasks. Some examples of robot or devices which have been made are:

Leonardo da Vinci designed and possibly built the first humanoid robot. The robot was designed to sit up, wave its arms, and move its head via a flexible neck while opening and closing its jaw.

Sony releases the first Aibo electronic dog in 1999 (Fig. 1).


Fig. 1

In 2002, Honda creates the Advanced Step in Innovative Mobility – ASIMO (Fig. 2). It is intended to be a personal assistant. It recognizes its owner's face, voice, and name. And also it can read email and is capable of streaming video from its camera to a PC.


Fig. 2

"TWENDY-ONE" (Fig. 3), a robot that coexists with humans, it features flexible movement of its joints and dexterous movement of its fingers.

Fig. 3

In consequence of using robots the concept of coexisting of humans and robots raised up. The definition of coexistence is:

1. Live close to each other without fighting.
2. The state of existing together at the same time or in the same place [4].
3. To live in peace with another or others despite differences, especially as a matter of policy [5].

From the moment of using such devices in ordinary life, human are getting much relevant to them day by day and the level of dependability in some cases are strongly high and in some industrial domains which are highly potent to human's damage risks using robots is actually inevitable. Application domains ask for human complement and substitution by robot. Some fields of applications are welding, painting, ironing, assembling, palletizing, product inspection, hazardous material handling or testing, and in technologies we may mention machine vision, end effectors design and artificial intelligence [6].

According to our research we can categorize robots into four major groups:

1. Personal robots
- Education/Hobbyist robots
- Entertainment robots like smart toys, robotic pets
- Partner robots
2. Industrial robots
- Agricultural robots
- Aerial robots like airframes, sensing and navigation
- Mine robots
3. Military robots
4. Space robots

In order to use robots in our daily life in this much vast scale safety and safe use of robots must be considered. Noticing that robots should make independent decisions, like move, run, stop, turn and so on, with their specific physical shape of the arms, sharp edges, material of the body, and heaviness, safety issues must be into specific attention in coexisting of human and robots. Remember the movie “The Modern Times” by Charlie Chaplin that a serving robot can may cause harm and injure human. There are many other similar examples about those incidents in today's world.

Specific robots, like entertainment or service robots, and maybe in future, partner robots, are in very close relation and coexistence with humans and unfortunately safety standards of human-robot relation are still not defined very well. Study in safety matters must be based on cybernetics, electronics, and mechanics and in psychology as well. In addition, safety standards should consider the different kinds of users for example children which may be in big danger to use specific kinds of robots.

Moreover, human behavior can also cause harm for robots. Misbehavior, improper use of a robot, bring up obstacles for moving robots, try to send meaningless or malicious codes to make it confuse are some examples of troubles that human can cause for robots which make chaos in coexisting of human and robots.

It seems that effective communication between human and robots strongly depend on the mutual understandings between the both sides (if there are understanding or learning cycles on robot sides). Three Asimov’s laws are a kind of approaches to make some rules and standard for the mutual existing domain although some studies show that these laws are not that much applicable to today’s world. In later part of the report, we will talk about the three laws and their proposed alternatives today.

## 2.3 Safety and Liveness Properties

### 2.3.1 Asimov’s Three Laws

Since their codification in 1947 in the collection of short stories I, Robot, Isaac Asimov’s three laws of robotics have been a staple of science fiction. Most of the stories assumed that the robot had complex perception and reasoning skills equivalent to a child and that robots were subservient to humans. In most situations, although the robots usually behaved “logically,” they often failed to do the “right” thing, typically because the particular context of application required subtle adjustments of judgment on the part of the robot. For example, the robot may be confused when determine which of the three Asimov’s laws take priority in a given situation.

The three laws have been so successfully inculcated into the public consciousness through entertainment that they now appear to shape society's expectations about how robots should act around humans. For instance, the media frequently refer to human robot interaction in terms of the three laws. They have been the subject of serious blogs, events, and even scientific publications. The Singularity Institute organized an event and Website, "Three Laws Unsafe," to try to counter public expectations of robots in the wake of the movie I, Robot, Both the philosophy[7] and AI[8] communities have discussed ethical considerations of robots in society using the three laws as a reference, with a recent discussion in IEEE Intelligent Systems [9]. Even medical doctors have considered robotic surgery in the context of the three laws [10].With few notable exceptions [11] [12], there has been relatively little discussion of whether robots, now or in the near future, will have sufficient perceptual and reasoning capabilities to actually follow the laws, And there appears to be even less serious discussion as to whether the laws are actually viable as a framework for human-robot interaction, outside of cultural expectations.

When it comes to details, Asimov's three laws were usually dealt with some basic safety standards in the past.

The first law of Asimov's laws is "A robot may not injure a human being or, through inaction, allow a human being to come to harm", but the biggest problem with it is that it views safety only in terms of the robot – that is, the robot is the responsible safety agent in all matters of human robot interaction.

The second law is "A robot must obey orders given to it by human beings, except where such orders would conflict with the first law". What's more interesting about the second law from a human-robot interaction standpoint is that as its core, it almost captures the more important idea that intelligent robots should notice and take stock of humans and that the people robots encounter or interact with can notice relevant aspects of robots' behaviour [13].

The third law is "A robot must protect its own existence as long as such protection does not conflict with the first or second law". The confusion, or puzzling about today's limited attempts to conform to the third law is that there are well-established technological solutions for basic robot survival activities that work for autonomous and human-controlled robots.

So we can find that when we try to apply Asimov's laws to today's robots, we may immediately run into problems, and find they are not so applicable to today's robots. Hence it is necessary to find a set of new laws which are typically suitable to today's world. According to Robin R. Murphy and David D. Woods discussion, there may be three alternative laws corresponding to Asimov's laws [14]. These three laws separately are "A human may not deploy a robot without the human-robot work system meeting the highest legal and professional standards of safety and ethics", "A robot must respond to humans as appropriate for their roles", and "A robot must be endowed with sufficient situated autonomy to protect its own existence as long as such protection provides smooth transfer of control to other agents consistent the first and second laws”.

According to the three Asimov’s laws as well as the alternative ones, we now have some idea about how the robots’ reflexes should be in order to retain the safety features. Then the next part is going to talk about what “hazard” is, and after that, a clear understanding of dangerous behavior can be obtained.

### 2.3.2 Hazards and Dangerous Behavior

In this project, we are mainly dealing with the service robot, to be specific, the drink servant robot.

In daily life, service robots are responsible for doing private tasks like nursing, house cleaning, security, life support and entertainment. For example “COBOT” and “TWENDY-ONE”, which are mentioned in the previous section, are some kinds of service robots that perform defined tasks in collaboration with human as assistance. The service robots are the robots that coexist with the human in our daily life. In our daily works and business we can have support from such types of robot.

When the coexistence of human and robot comes to consideration, the safety and liveness issues also come in parallel. As mentioned before, although there are many benefits that we can get from industrial and service robots, they can be a great threat for the human’s safety. If there are no safeguards in the industrial environment, then the industrial robots can be responsible for creating dangerous conditions. For example, in 1981, a 37 years’ old factory worker named Kenji Urada entered a restricted safety zone at a Kawasaki manufacturing plant to perform some maintenance on a robot. In his haste, he failed to completely shut down the unit. Then the robot’s powerful hydraulic arm pushed the engineer into some adjacent machinery, thus making Urada the first recorded victim to die at the hands of a robot [15]. This example clearly supports Morita et al.’s observation that when task-performing robots and humans share the same physical space, the overriding goal must be to ensure human safety [16].

There are basically three potential hazards associated with robotic systems which are as follows:

1. Impact — this involves such things as being struck by a moving part of the robot, or by parts or tool carried or manipulated by the robot. It can be caused by the unexpected movement of the robot or by the robot ejecting or dropping work pieces or molten metal.

2. Trapping — this can be caused by the movement of the robot in close proximity

to fixed objects like machines, equipment, fences, etc. Trapping points can also be caused by the movement of the work carriages, pallets, shuttles or other transfer mechanisms. They can also be presented on the robot itself on the arm or mechanism of the robot.

3. Other — this would include hazards inherent to the application itself like electric shock, arc flash, burns, fume, radiation, toxic substances, noise, etc. [17]. These hazards can arise from several sources and should be considered in typical robot installations which include:

- Control Errors
- Mechanical Hazards
- Environmental Hazards
- Human Errors
- Ancillary Equipment

It is clear that during the interaction between the user and our service robot, all these kinds of hazards may happen, and what we need to do first is to make every effort to assure the safety of the interaction.

### 2.3.3 Safety and Liveness Properties

However, although it is said in the previous section that we would like to make every effort to assure the safety of the drink servant robot, usually there are always two kinds of properties we want the robot to satisfy, besides safety properties the other is liveness properties. Their definitions are listed below [18]:

- Safety properties, which state that something bad never happens – that is, that the program never enters an unacceptable state.
- Liveness properties, which state that something good eventually does happen – that is, that the program eventually enters a desirable state.

According to our understanding, “safety” here means robots operate within their spec and within safe limits. The factors that may have influence on robot safety can be divided into two parts: the human factors and the robot factors [19].

As for the human factors, many of them, such as the layout of the robot control panels, the materials of the robot used, the design of safety barriers as well as teach-pendants training for personnel who operate or service the robot, all have a direct and obvious influence on safety. One important consideration is, in order to keep the human safe in human robot interaction, how the robot should act indeed. For example, for our coffee servant robot, the design of the robot’s moving speed is a problem that should be taken into carefully consideration as too fast a speed may cause the coffee spill out of the cup, thus leading to unexpected safety problems. Another important human factors area concerns how humans should be alerted to potentially dangerous situations involving robots. Although the use of warning signs, audible alarms and flashing lights is discussed in many of the previously and recent works, there does not seem to a consensus on exactly where and how they should be deployed.

As for the robot factors, they usually come from the design and implementation of the robot itself, and also its related support equipment. Of all these factors, the design and implementation factors may involve the design of implementation of all or any parts of the robot, such as warning devices, barriers, interlocks and even a physical part like, the “arm”! Moreover, robot safety equipments can roughly be classified into two groups [20]:

1. The equipment which acts to prevent humans from coming into the workplace of an activated robot
2. The equipment that is used to detect humans within the robot workspace

These equipments include not only physical barriers like fences and partitions, but also devices such as photoelectric light-beam curtains or capacitive fields that are interlocked to the robot controller in such a manner that crossing into the robot workspace disables the robot.

Besides safety properties there are still another kind of properties we want the service robot to satisfy – liveness properties.

The only liveness property that has received careful formal treatment up till now is program termination. However, concurrent programs are capable of many more sins of omission than just failure to terminate. Indeed, for many concurrent programs – operating systems are a prime example – termination is known by the less flattering name of “crashing”, and we want to prove that it does not happen. For such programs other kinds of liveness properties are important, for example:

- Each request for service will eventually be answered.
- A message will eventually reach its destination.
- A process will eventually enter its critical section.

For a sufficient understanding of the two kinds of properties, let us take the traffic lights as an example. In a crossroads, we need not only to guarantee the traffic lights are working, for instance, they won’t stop working due to lack of power, but also to ensure the traffic lights work properly, for instance, no signals for two adjacent directions are the same at a time, i.e., they are both showing red – which is a waste of resources – or green – which may even cause traffic accidents!

From the example above, we can see that liveness properties are as important as safety ones. However up till now, although a number of methods have already been proposed for proving safety properties, formal proof of liveness properties has received little attention, which is also another reason for this project.

Up till now, it has already talked about dangerous problems, safety problems and liveness problems, and to some extend a basic understanding of their potential solutions has also been provided. The next part it is going to talk about safety reflex of the robot, in other words, what the robot should react in some given situations while not violate the safety requirements.

### 2.3.4 Safety Reflex

A reflex action, also known as a reflex, according to the explanation, is “an involuntary and nearly instantaneous movement in response to a stimulus [21]”. In most contexts, in particular those involving humans, reflex actions are mediated via the reflex arc. Although this is not always true in other animals, nor does it apply to casual uses of the term “reflex”, it can still be recognized that there exists a reflex action in the interaction between the service robot and human.

The control diagrams of many service robots include hardware- or software-based collision prevention modules that detect and avert collision situations. The procedure includes monitoring the distance between the user and the robotic arm, measuring the space between the user and the object fixed in the gripper, checking for dangerous proximity between the payload and other links of the robot, and calculating the distance between the robot and nearly located obstacles [22]. Such a system is intended to stop the robot in case of potential danger of collision. However, even if an imminent collision is detected, the robot may still harm the human due to the inertia of the arm or a delay from the detection signal to the activation of the stopping mechanisms. Robots must be safe to users even when their main controller fails, causing no harm or minimal harm to the user.

To solve the problem, we need to propose some kind of a mechanism, in which the action is quite similar to the human reflex behavior. As a response to unintentionally touching a very hot object, one instinctively jerks his hand back. The safety reflex to be developed follows the same behavior pattern and for example immediately moves the robotic arm back when it collides with the user. This way, the forces applied to the user cannot reach critical values and the pressure impact on the skin is minimized. In order to make the mechanism effective even in cases where the main controller fails, Noriyuki Tejima and Dimitar Stefanov has proposed an idea of separating powering and control of the reflex mechanism [23].

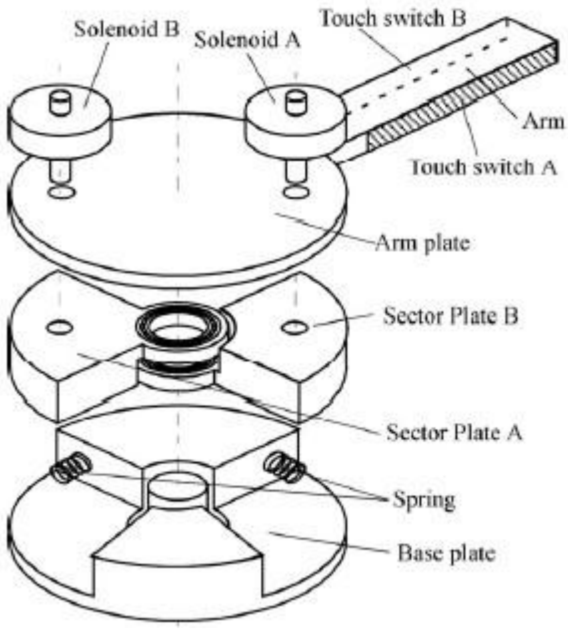



Fig. 4 [23]

The structure of such a reflex mechanism is shown in Fig.4. And the reflex mechanism transmits motor power to a rotating joint of the robot. The sector plate consists of two components, noted as sector plate A and sector plate B, respectively. Each component of the sector plate is connected to the arm plate via a pin that is linked to the plunger of a push-type solenoid (solenoid A and solenoid B). While energized, the solenoids keep the pins between the plates and do not allow their mutual rotation. Both segments of the sector plate are linked to the base plate via a central shaft fixed to the base plate. Two compressed coil springs keep the head surfaces of these segments a certain distance from the base plate. The actuator is also connected to the base plate. The robotic link is allied to the arm plate.

Here the proposed mechanism is just like the design of the robot arm, and the mechanism makes it quite like a real human arm. However the physical design is not the focus point in this project. In fact what we are interested in is more like, “if the user does something in the human-robot interaction, the robot should do what kind of things corresponding to it”, or what mechanism should be built in the drink servant robot to make sure the interaction safe and live enough, which is obviously in first-order logic. So the mention of the physical reflex mechanism is only for the purpose to enhance the understanding of the concept of “reflex” as well as safety reflexes.

## 2.4 Related Work and State-of-the-Art

### 2.4.1 Safety Principle for Service Robot

The basis for safety in a robot cells comes from a European standard “DIN EN 775

Safety of manipulating robot [24]", which has now been converted to an international standard "ISO 10218: Robots for industrial environments – Safety requirements". The possible installation of a robot system within reach of a human was considered under highest conditions in DIN EN 775, and one target of the ISO 10218 now is to further provide regulations for robot-human cooperation [25].

Nowadays, most companies in the United States adhere to the ANSI/RIA R15.06-1999 safety standard. While this standard is not required legally, OSHA does refer to it when addressing robots. The ANSI/RIA safety standards provide specific safety criteria about which devices and measurements to implement for each robot configuration [26].

Besides ANSI/RIA, a more famous organization for standardization is ISO/TC184/SC2. So far the committee has developed 11 international standards and 4 technical reports, and some of the most important standards are in the field of safety, performance criteria and interfaces for mechanics and software. And a good example of a global approach is the ongoing revision of the safety standard for robots [27].

The first occasion to carry out the certification of service robot by "NPO the Safety Engineering Laboratory", which was founded by safety professionals in 2002 in Tokyo, was at Aichi EXPO 2005 in Japan, where approximately 100 various kinds of service robots in developing phases were demonstrated during the EXPO from June to September 2005. To keep the safety of exhibited service robots at AICHI EXPO, a research committee on safety guideline of robots had been established and an adequate safety guideline had been set up. Here the well-tried international standards on safety, e.g. ISO/IEC Guide 51 guideline for the inclusion of safety aspect in standards [28], ISO12100 – the general principle of safety of machinery [29], ISO14121 - the principle of risk assessment [30] etc. were adopted as concept to ensure the safety of service robot for the EXPO. Particularly the inherent safety design on the base of ISO/IEC Guide 51 and also the safety management including documentation and communication were found essential. No accident caused by the exhibited service robots had been reported during the period of AICHI EXPO. The basic procedure, well-practiced for AICHI EXPO, has been and will be adopted for various kinds of service robots as safety principle further on in Japan.

### 2.4.2 Prior Art: CHRIS Project at BRL

The CHRIS, which stands for Cooperative Human Robot Interaction System, is a project will address the fundamental issues which would enable safe Human Robot Interaction (HRI). Specifically this project addresses the problem of a human and a robot performing co-operative tasks in a co-located space, such as in the kitchen where your service robot stirs the soup as you add the cream. These issues include communication of a shared goal verbally and through gesture, perception and understanding of intention from dexterous and gross movements, cognition necessary for interaction, and active and passive compliance. These are the prerequisites for many applications in service robotics and through research will provide the scientific foundations for engineering cognitive systems.

The project is based on the essential premise that it will be ultimately beneficial to our socioeconomic welfare to generate service robots capable of safe co-operative physical interaction with humans. The key hypothesis is that safe interaction between human and robot can be engineered physically and cognitively for joint physical tasks requiring co-operative manipulation of real world objects. A diverse set of disciplines have been brought together to realize an inter-disciplinary solution.

The starting point for understanding cooperative cognition will be from the basic building blocks of initial interactions, those of young children. Engineering principles of safe movement and dexterity will be explored on the three available robot platforms, and developed with principles of language, communication and decisional action planning where the robot reasons explicitly with its human partner. Integration of cognition for safe co-operation in the same physical space will provide significant advancement in the area, and a step towards service robots in society [31].

The project has partners like Bristol Robotics Laboratory in UK, Centre National Recherche Scientifique in France, Fondazione Istituto Italiano di Technologia in Italy, and Max Planck Gesellschaft zur Forderung der Wwissenschaften e.V. in Germany. And our study is carried out just at the Chris Project area at BRL, using equipments and devices of that project like Human Eye and Gaze Tracking.

The CHRIS architecture is a cognitive architecture which is designed as platform independent (Fig). And in our project, parts of them are used to fulfill our needs, like the egoSphere and the 3D Perception, which will be mentioned in a later part.

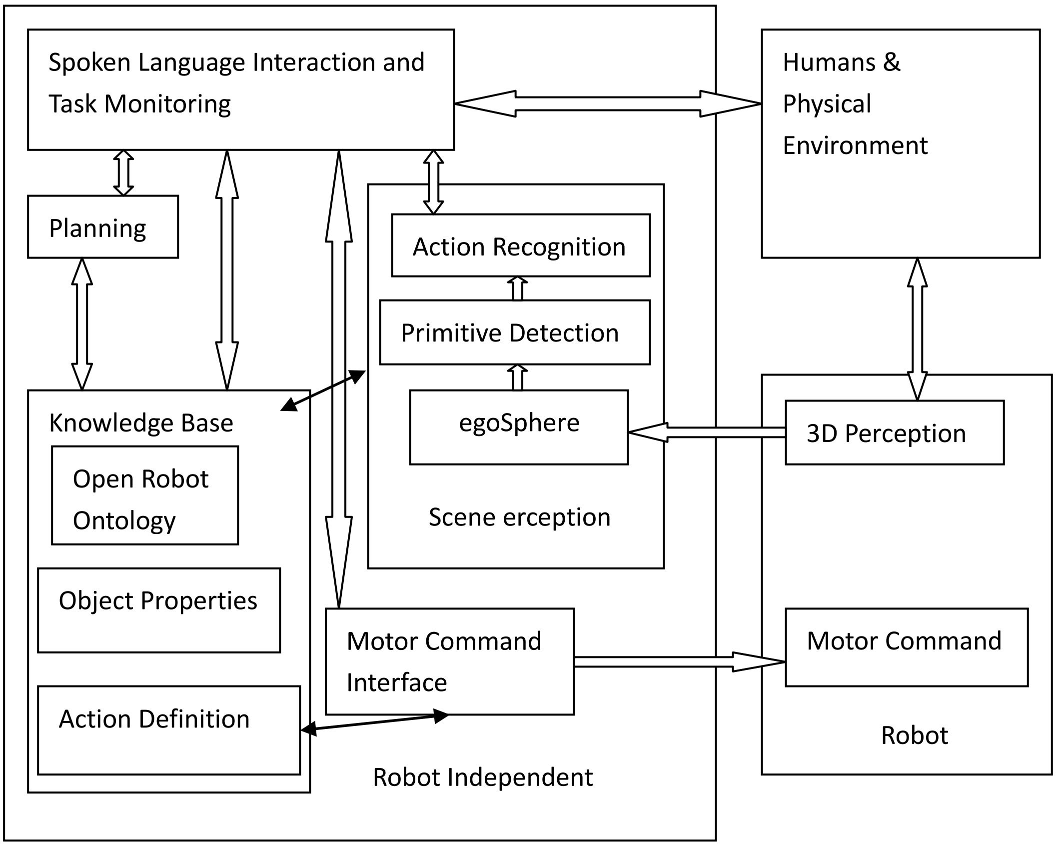



Fig. 5[32]

### 2.4.3 Prior Art: Object Provider

The Object Provider is a module widely used at Bristol Robotics Lab, aiming to provide the information of the objects in the egoSphere, which will be introduced in a later part. With the help of the cameras as well as the knowledge base, it can be used to locate the position, detect the movements alongside with showing other description of the objects, for example, their ID.

Corina Grigore modified the details of the Object Provider, so that it can directly show the object ID in the knowledge base, the object ID in egoSphere and the movements like yawn, pitch and roll. Thus it can be modified to meet our needs. A detailed description of the Object Provider, which to be specific are the cup and hand wrist in this project, will be given in Chapter 4.

## 2.5 Summary

This section has introduced the background knowledge of robots, including the current coexistence status of the robot and the human; besides, it also has illustrates the concepts of safety, liveness and reflex, and as well as the work we mainly refer to, for example, the Chris Project, which focuses on the cooperative human robot interaction. Then we now have a basic understanding of the issues we try to address in our project. Safety is the issue we need to assure in all the cases, while liveness is the property that we want the robot perform more reliable and acceptable. Safety reflex is the reaction of the robot in particular situations while it does not violate the requirements of safety at the same time.

In next section, it is going to talk about specification of this project.

# Chapter 3: Project Specification

## 3.1 Introduction

This chapter aims to show the detailed specification of the project. Firstly, what the project intends to do will be illustrated in the specification part. Description of the context and the object is next given, which in the project is a drink servant robot's behavior. Finally the concept of behavioral adaptation is demonstrated and some potential behavioral adaptations during the process of handing over a drink are listed, although they are not exactly the same with that in our selected scenarios, which will be introduced in a later part.

## 3.2 Specification

This project is designed to explore potential safety reflexes in human-robot interaction, especially in those human-assist robots. Issues to be investigated are both safety properties and liveness properties, in other words, safety reflexes need to comply with liveness requirements as much as possible.

In a nutshell, the compromises between safety values and liveness values are expected to be found out, which can be recognized as an approach applicable to the service robots. But first of all, latent hazards of these human assist robots should be analyzed, and their corresponding solutions, which are also known as safety reflexes or safe behavioral adaptations, should be figured out as well.

In service robots, as mentioned previously, hazards may come from many different aspects, including machinery itself and the system design, where the latter part is what this project concerns. In real industrial and academia quarters, every designing possibility should be studied. It may cost a lot of time, so does in our project. As a result, in order to study safety reflexes whilst not to spend too much time on identifying only potential hazards, the scope of service robots and the scenarios should be narrowed down, thus these safety and liveness requirements can be concretized and validated.

## 3.3 Context and Object

Then, as the title of the dissertation pertains to, the context of this project is adaptable robot behavior, to be specific, the action of the robot when serving a drink to the human.

According to common knowledge, behavioral adaptations are things organisms do to survive. Consider an animal in the wild which needs to compete with other members of its own species for limited resources while meanwhile needing to adapt itself to the environment we can probably suppose that it must have a strong ability to survive. This kind of ability not only includes the ability to win the competition with its fellows for resources, but the ability to acquire food, the ability to escape from its predator and the ability to adapt itself to the weather and other environmental substances as well, and in other words, it is a kind of ability to solve conflicts. Here, abilities or behaviors are called adaptable behaviors.

In general machine learning terms, a behavior is a hypothesis that classifies different stimuli into a set of actions, and the adaptation changes to a behavioral hypothesis based on the observed examples. In brief, the behavioral adaptations are merely the corresponding reactions.

Robot study is a branch of Artificial Intelligence which aims to simulate and model the intelligence of animals, including humans. In this project, the subject is a drink servant robot, and thus there is an assumption that it should have the same or similar behaviors as a real servant.

Hence, the project will focus on the studies of those reactions and reflexes in the context of a drink servant robot, to be specific, in the scenarios where the drink servant robot serves a drink to a human. These different scenarios will be introduced at length in the following section.

As mentioned previously, the object of study will be a drink servant robot with a head that is capable of wide-field in-scene head location, head-direction and gaze-checking, although only the last function will be used in the implementation and experiments; besides, it also includes an associated robot arm with grip. Then, detailed behavioral adaptations will be mentioned in the next section.

## 3.4 Behavioral Adaptations

In the context of a drink servant robot, the first issue worthy of consideration is the safety properties, which means that it must be safe enough during the entire process the robot serves the drink to the human. In addition, the issue concerning the liveness properties is also paid attention to. Hence, the behavioral adaptations of our drink servant robot should comply with these two aspects of the requirements.

Here, the main potential situations which may happen in our scenarios and which require special attention are considered, and the possible "triggers" of safety reflexes and dangerous adaptations are deciphered along with acceptable compromise adaptations (Table1).

Table1

| | Safety adaptations | Dangerous adaptations | Acceptable (compromised with safety and liveness |
|---|---|---|---|

| | | | |
|---|---|---|---|
| | | | properties) adaptations |
| Gaze-checking (user not looking) | 1.Stop the "handing over" action;<br>2.Return to the initial position. | Do the "handing over" action, and the drink may be chucked over at the human | Stop the "handing over" action and do something to remind the user to look at the cup, i.e. say something like "please look at the cup". |
| Grab-checking (user not grabbing) | 1.Stop the "handing over" action;<br>2.Return to the initial position. | Do the "handing over" action, and the drink may be chucked over at the human | Stop the "handing over" action and do something to remind the user to grab the cup, i.e. say something like "please grab the cup". |
| The robot asks "do you feel like a drink" and waits for the human's answer | Have only two answers "yes" and "no", but it is not liveness as maybe at first the human does not want the drink while sometime later he maybe like a drink. | | Have "yes", "no" and "exit" three answers, which corresponds to "serve a drink", "wait sometime and ask again" and "stop working" separately. |
| The robot asks "are you sure you have the cup" and waits for the human's answer | When the human answers "no", the robot stops working and moves to the initial position directly. | When the human answers "no", the robot still does the "handing over" action | When the human answers "no", the robot asks the user to grab the cup again. |

## 3.5 Summary

This chapter mainly focuses on the specification of this project. It is intended to identify the safety and liveness requirements in human robot interaction, however, in order to test and validate, these requirements are concretized to detailed ones in a specified scenario, which is the drink servant robot serves the drink to the human.

The next chapter is the design and implementation phase, which is also the main part of this project and has totally been done at Bristol Robotics Laboratory. It consists of the selection and specification of detailed scenarios, and the design and implementation of those scenarios as well.

# Chapter 4: Project Design

## 4.1 Introduction

The main idea of this chapter is to introduce how the system, or in other words, how the drink servant robot works in our project. To begin with the selected scenarios of the study is provided, and then the state diagram which explains how the system works is shown, and finally a detailed description of each component is listed to give a clearer understanding of the design of the project.

## 4.2 Scenarios

Due to different safety requirements, there are mainly two scenarios in this project, one for water and the other for coffee. Obviously this requires a higher safety standard when the drink servant robot serves drinks like coffee compared with that of water. When a cup of water is spilled over at the user, it may not cause that much damage as the clothes will look normal again as soon as is they have dried; however the situation may be completely different if a cup of coffee is spilled over at the user, even though the clothes can be clean again with the help of detergent or other similar things, which will ultimately drain energy, or even money. As a result, it was decided to design three steps to ensure safety during the coffee scenario, while ensuring safety during the water scenario took only one.

### 4.2.1 Water Scenario

1. Robot initializes itself, and says “Initializing myself; Initializing OK”;
2. Robot waits command from the terminal;
3. User types “start” in the terminal;
4. Robot says “Hello, I’m Bert 2, do you feel like a drink”;
5. User answers “Yes”;
6. Robot says “Water or coffee”;
7. User answers “Water”;
8. Robot says “Preparing the drink”;
9. Robot’s grip opens;
10. Robot’s grip closes;
11. Robot’s hand move up to the serve position;
12. Robot says “Please grab the cup”; (totally for 3 times)
13. User looks at the cup and at the same time grab the cup;
14. Robot says “Enjoy”;
15. Robot’s grip opens;
16. User takes over the cup;
17. Robot waits for 20 seconds;
18. Robot says “Do you like another drink”;
19. User answers “Yes”;

20. Robot says “Water or coffee”;
Etc…
5a. If User answers “No” then the system goes to step 17;
5b. If User answers “Exit” then Robot says “Goodbye” and stops working;
13a. If User does either not look at the cup or not grab the cup then the system goes back to step 12;
13b. If User neither grabs the cup nor look at the cup then the system goes back to step 12;
19a. If User answers “No” then the system goes back to step 17;
19b. If User answers “Exit” then Robot says “Goodbye” and the system stops working;

### 4.2.2 Coffee Scenario

1. Robot initializes itself, and says “Initializing myself; Initializing OK”;
2. Robot waits command from the terminal;
3. User types “start” in the terminal;
4. Robot says “Hello, I’m Bert 2, do you feel like a drink”;
5. User answers “Yes”;
6. Robot says “Water or coffee”;
7. User answers “Coffee”;
8. Robot says “Preparing the drink”;
9. Robot’s grip opens;
10. Robot’s grip closes;
11. Robot’s hand move up to the serve position;
12. Robot says “Please pay attention to the cup”; (totally for 3 times)
13. User looks at the cup;
14. Robot says “Please grab the cup”; (totally for 3 times)
15. User looks at the cup and at the same time grab the cup;
16. Robot says “Are you sure you have the cup”;
17 User answers “Yes”;
18. Robot says “Enjoy”;
19. Robot’s grip opens;
20. User takes over the cup;
21. Robot waits for 20 seconds;
22. Robot says “Do you like another drink”;
23. User answers “Yes”;
24. Robot says “Water or coffee”;
Etc…
5a. If User answers “No” then the system goes to step 21;
5b. If User answers “Exit” then Robot says “Goodbye” and stops working;
13a. If User does not look at the cup, then the system goes back to step 12;
15a. If User looks at the cup but does not grab the cup, then the system goes back to step 14;

15b. If User grabs the cup but does not look at the cup, then the system goes back to step 12;
15c. If User neither look at the cup nor grab the cup, then the system goes back to step 12;
17a. If User answers “No”, then the system goes back to step 14;
22a. If User answers “No”, then the system goes back to step 21;
22b. If User answers “Exit” then Robot says “Goodbye” and the system stops working;

## 4.3 System Overview

In the system, there are four modules in total (Fig. 6), among which the Framework is the core as it is used as the bridge and scheduler.

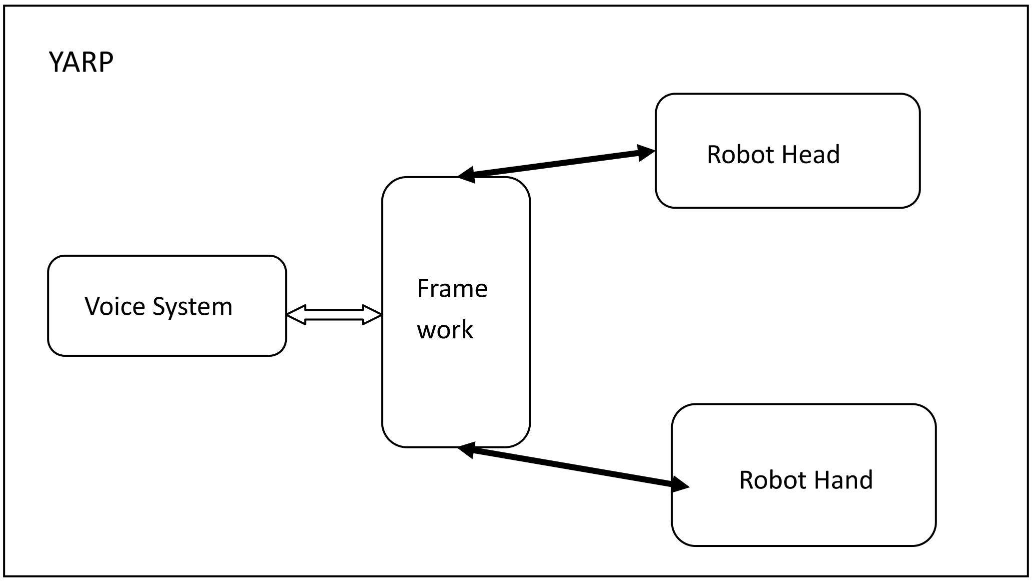



Fig. 6

### 4.3.1 Framework

As shown above, the “Framework” is the core module in our program, and acts as a connecter and a controller for the whole job. The importance of Framework is mainly illustrated in the following aspects.

First, many small functions which complete certain tasks respectively are written in the Framework. For example, the one named “moveForward” is used to control the robot hand (Fig. 7), together with the help of the function “writeSerial”(Fig. 8), which is designed to be written into the Serial Port. The two parameters in moveForward are the corresponding channel for controlling the robot hand to move up and down and the position to serve the drink separately, while SPEED is the value which stands for the moving speed and is predefined in the program. Additionally, there are six different channels in the robot hand, where 0 is the channel to control the arm movement, and 4 is the channel to control the grip for opening and closing.

```
bool Framework1::moveForward(int channel, int position)
{
    int noOfChars = 0;
    char sendData[100];

    noOfChars = sprintf_s(sendData, 100, "#%d P%d S%d \r", channel, position, SPEED);
    cout<<sendData<<endl;
    for(int i=0; i<noOfChars;i++)
        if(!writeSerial(sendData[i]))
            return false;

    return true;
}
```

Fig. 7

```
bool Framework1::writeSerial(char charr)
{
    DWORD dwNumBytesWritten;

    WriteFile (hPort,              // Port handle
               &charr,              // Pointer to the data to write
               1,                  // Number of bytes to write
               &dwNumBytesWritten, // Pointer to the number of bytes
                                 // written
               NULL                // Must be NULL for Windows CE
    );

    return true;
}
```

Fig. 8

Moreover, Framework is the dispatcher and scheduler for the whole system. It has a function named “updateModule”, which controls the whole process by assigning different states which allows the whole system to work properly. Furthermore, the “open” function sets up all the connections for input and output of different modules (Fig. 9).

```
bool Framework3::open()
{
    // setting up connection with voice system for input
    if (!portI1.open(fmI1.c_str()))
        cout<<"Problem with port for voice system input."<<endl;
    if (!yarp::os::Network::connect(rpcO.c_str(), fmI1.c_str()))
        cout<<"Problem with connection for voice system input."<<endl;
    //setting up connection with terminal for input
    if (!portI2.open(fmI2.c_str()))
        cout<<"Problem with port for terminal output."<<endl;
    if (!yarp::os::Network::connect(terminalO.c_str(), fmI2.c_str()))
        cout<<"Problem with connection for terminal output."<<endl;
    //setting up connection with gaze module for input
    if (!portI3.open(fmI3.c_str()))
        cout<<"Problem with port for gaze input."<<endl;
    if (!yarp::os::Network::connect(gazeO.c_str(), fmI3.c_str()))
        cout<<"Problem with connection for gaze input."<<endl;
    //setting up connection with hand wrist module for input
    if (!portI4.open(fmI4.c_str()))
        cout<<"Problem with port for hand wrist input."<<endl;
    if (!yarp::os::Network::connect(hrO.c_str(), fmI4.c_str()))
        cout<<"Problem with connection for hand wrist input."<<endl;
    //setting up connection with voice system for output
    if (!portO1.open(fmO1.c_str()))
        cout<<"Problem with port for voice system output."<<endl;
    if (!yarp::os::Network::connect(fmO1.c_str(),rpcI.c_str()))
        cout<<"Problem with connection for voice system output."<<endl;
    //setting up connection with gaze module for output
    if (!portO3.open(fmO3.c_str()))
        cout<<"Problem with port for gaze output."<<endl;
    if (!yarp::os::Network::connect(fmO3.c_str(), gazeI.c_str()))
        cout<<"Problem with connection for gaze output."<<endl;
    //setting up connection with hand wrist module for output
    if (!portO4.open(fmO4.c_str()))
        cout<<"Problem with port for hand wrist output."<<endl;
    if (!yarp::os::Network::connect(fmO4.c_str(), hrI.c_str()))
        cout<<"Problem with connection for hand wrist output."<<endl;
    return true;
}
```

Fig. 9

### 4.3.2 State Diagram

Excluding the start state and the end state, there are ten states in total in the program (Fig. 10). These states are used to control the actions of the system. This saves much time in using different states to represent different system responses, for example, when the state reaches "AnotherCup", it actually consists of several different origins. To be specific, when the voice system asks "Hello I'm Bert 2, do you feel like a drink", the system state is "AnotherCup"; and also when the voice system asks "Would you like another drink", the system state is still "AnotherCup". This way to control the system was chosen for the reason that, although the voice system asks different words, the following actions and responses of the system are absolutely the same. Further to this, the states also make the system look less complex.

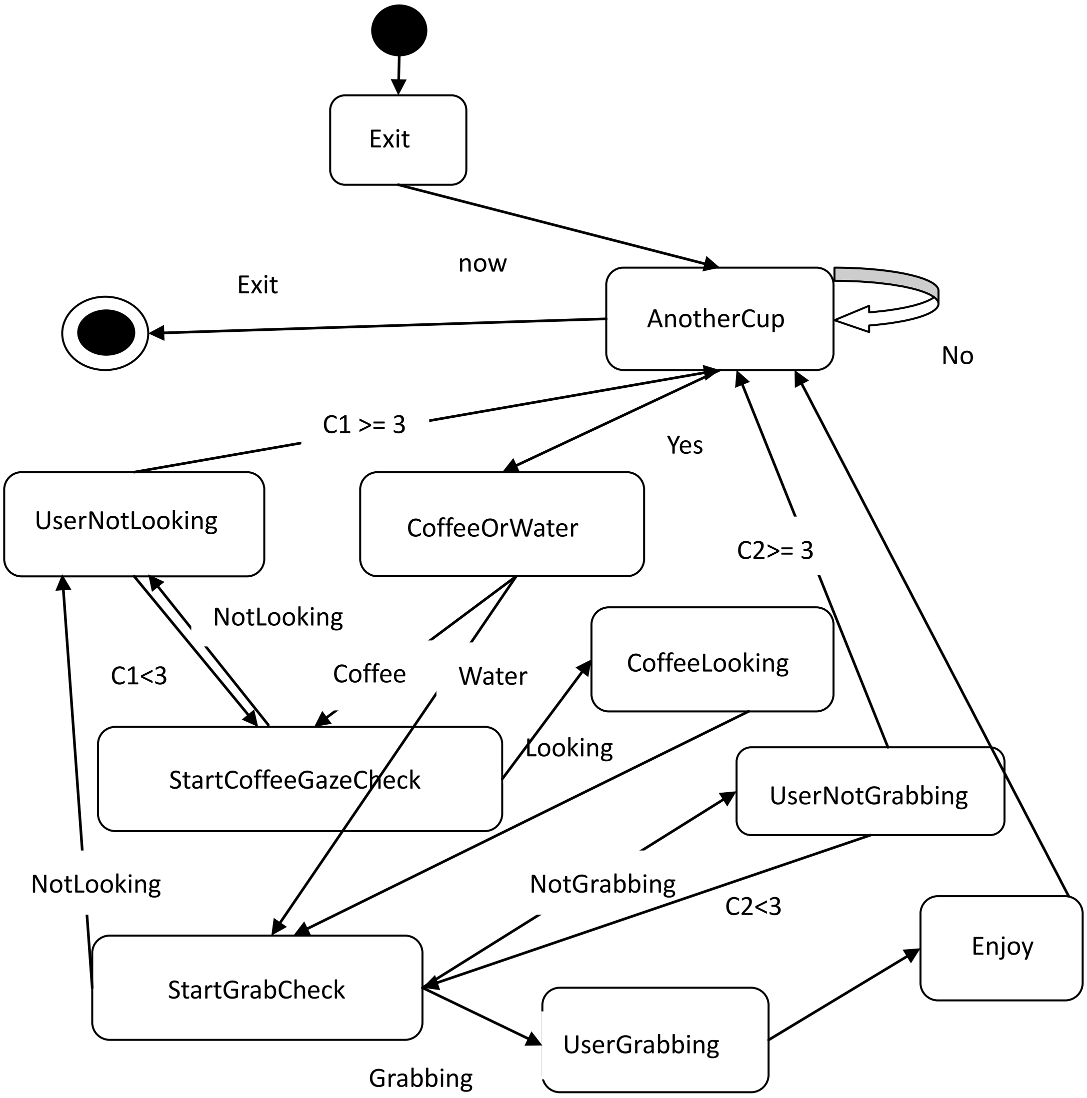



Fig. 10

## 4.4 Components/ Modules

### 4.4.1 YARP Platform

YARP stands for Yet Another Robot Platform, and is a set of libraries, protocols, and tools to ensure modules and devices are cleanly decoupled. It is reluctant middleware, while it is definitely not an operating system. If the data is the bloodstream of the robot then YARP is the circulatory system. More specifically, YARP supports building a robot control system as a collection of programs communicating in a peer-to-peer way, with a family of connection types like TCP, UDP and so on. These connection types can be swapped in and out to meet the needs of the researchers. Besides, YARP also supports similarly flexible interfacing with hardware devices.

The components of YARP can be mainly divided into three parts: libYARP_OS which interfaces with the operating system(s) to support easy streaming of data across many threads across machines; libYARP_sig, which performs common signal processing tasks in an open manner, can be easily interfaced with other commonly used libraries, for example OpenCV; and libYARP_dev which interfaces with common devices used in robotics. These components are maintained separately, and the core component is libYARP_OS, which must be available before the other components can be used.

In this project, many of the modules and hardware devices that have been mentioned in previous chapters, such as the robot hand and the vicon system, are involved, and require synchronization in order to get them to work simultaneously or sequentially. As the YARP model of communication is transport-neutral and the data is decoupled from the details of the underlying networks and protocols in use, and because YARP uses a methodology for interfacing with devices that encourages loose coupling and can make changes in devices less disruptive, it was decided to make the whole project under the YARP platform. By doing so, the complicated pile of hardware as well as the equally complicated pile of software could easily be controlled.

There are a total of four machines for this project, and each of them controls one or several different devices or modules (Fig. 11). Specifically speaking, one machine controls the YARP server, another controls the vicon system whose abstraction layer is the egoSphere, the third takes charge of the face lab which is used for gaze checking, while the last one finishes complicated tasks like controlling the robot hand and the voice system, running the framework and other involved programs.

The YARP server is used to find out where the other machines are, for instance, to translate port names into IP addresses and port numbers. The communication between software modules happens through a peer-to-peer format, for instance, between the machines. With these kinds of settings, the modules and devices involved less coupled. Furthermore, the data needed could be streamed to access different parts of the project connected and conveniently communicate with each other. For example, if a list of numbers on terminal 1 were typed (“yarp wirte”), they would show up on terminal 2 (“yarp read”), and the network is as simple as the figure below shows (Fig. 12). Finally, YARP is written in C++, so it is normally used as a library in C++ code while any application that has a TCP/IP interface can talk to YARP modules using a standard data format.

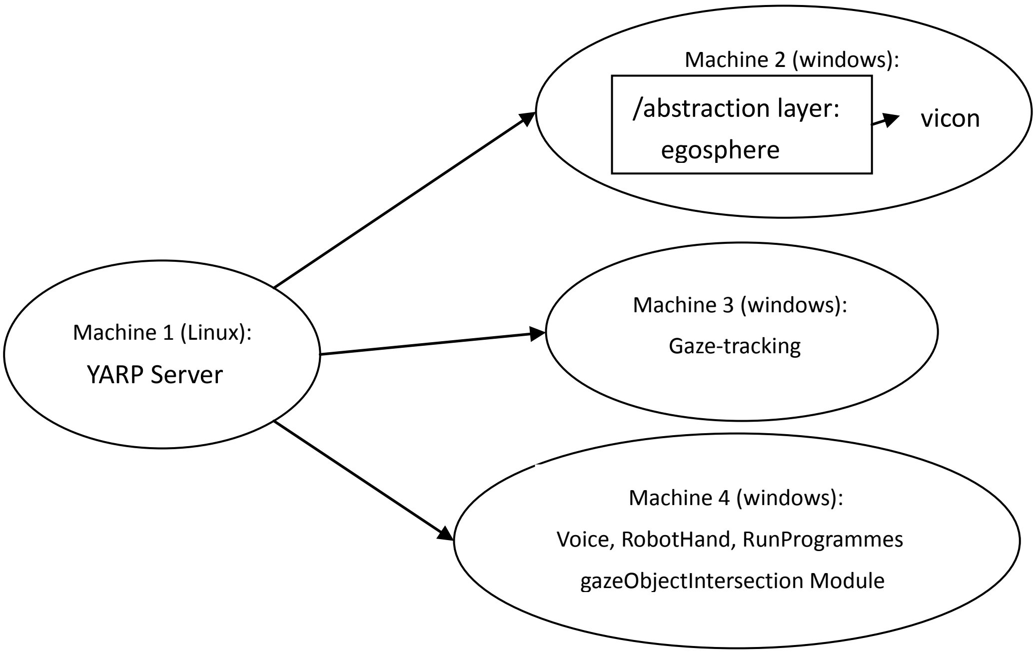



Fig. 11

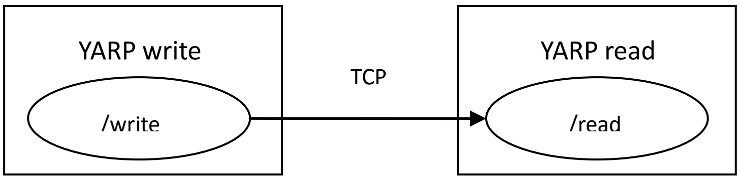



Fig. 12

### 4.4.2 Vicon System/ EgoSphere

The first layer of abstraction between the sensory perception systems, and higher level cognitive architectures and motor control elements is formed by the egoSphere. In other words, we work with the abstraction layer egoSphere in our project, instead of using the vicon system directly.

The vicon system at Bristol Robotics Laboratory is a system with eight cameras placed in the roof of the system space (Fig. 13). With the help of the cameras and the markers, which are small round balls but can be detected by the cameras, information about the objects in the system space can then be retained and stored.

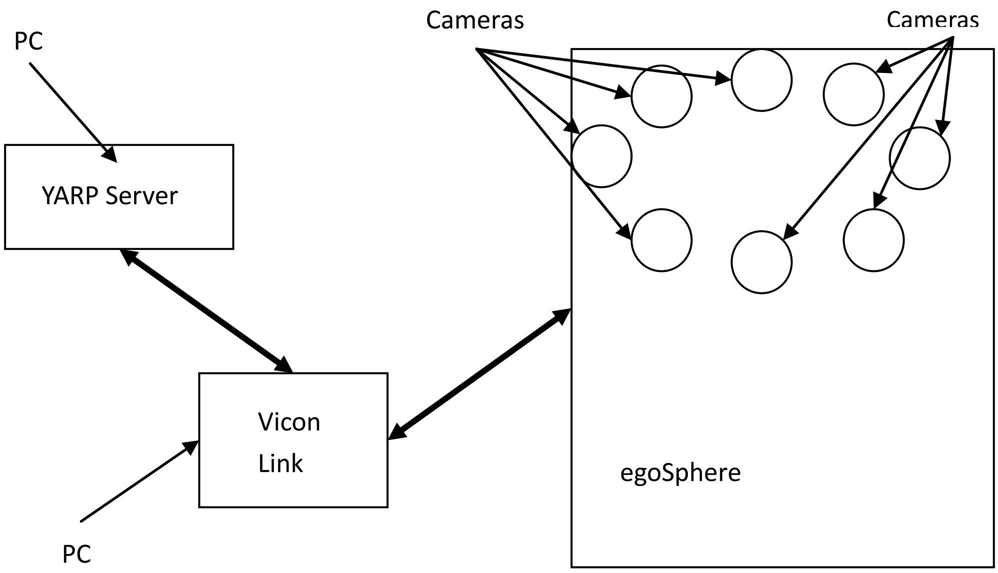



Fig. 13

The egoSphere used at Bristol Robotics Laboratory acts as a fast, dynamic, asynchronous storage of object positions and orientations. Here, the object positions are stored in spherical coordinates using radius, azimuth and elevation, and the object orientation is stored as rotations of the object reference frame around the three axes x, y and z, of a right-handed Cartesian world-frame system. The origin of the world frame can be chosen arbitrarily and, for this project and the experimental work, it was located 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.

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. No assumptions are made about the nature of an object and any further information like object name and object type will have to be queried from the Knowledge Base using the objectID. However, this kind of information is not what was focused on for this project and thus will not be dealt with in too many details in terms of this Knowledge Base. This kind of 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 egoShpere include a client class which provides methods like addObject(.), setObject(.), getObject(.) or getNumberOfObjects(.) and so on. It is clear that, in its current state, the egoSphere is merely a convenient abstraction layer.

In this project, as the egoSphere has already been deployed, the thing needed for the first step was to add the objects we may use into the egoSphere. First, 6 markers were placed on the hand wrist model and 4 markers on the cup. The positions of the markers were deliberately chosen, and by choosing those positions the wrist and the cup could be presented in the best way. As soon as there are markers on the objects, the hand wrist model and the cup could then be detected by the cameras and shown on the screen in a 3D way. Then, we added the objects into the egoSphere using the software vicon link, and after doing so, a unique egoSphere objectID was assigned to each object randomly.

In order to get information of these egoSphere objects, we have to know the egoSphere objectID of each object. We managed to do the task with help of its corresponding relationship with the real objectID, which acts as a shared key across to several databases, by writing codes in C++. The figure below (Fig. 14) shows the egoSphere objectID for all the objects involved in the project.

```
yarp: Port /framework1/voice:o active at tcp://164.11.116.64:10252
yarp: Sending output from /framework1/voice:o to /chris/radCommunication/framew
rk:i using tcp
yarp: Port /framework1/gaze:o active at tcp://164.11.116.64:10262
yarp: Sending output from /framework1/gaze:o to /gaze/framework1:i using tcp
yarp: Port /framework1/hr:o active at tcp://164.11.116.64:10272
yarp: Sending output from /framework1/hr:o to /hr/framework1:i using tcp
Connections established.
SERIAL PORT \.\\COM1 READY!
PORT PARAMETERS: 115200,n,8,1 [baudrate,parity,databits,startbits]

serial open succesful
yarp: Port /framework1/egosphere active at tcp://164.11.116.64:10292
yarp: Sending output from /framework1/egosphere to /chris/egosphere/rpc using t
p
cup: 5 hand wrist: 4 robot head: 0
 human head: 1 gaze ray: 2 gaze int point: 3
```

Fig. 14

## 4.4.3 Hand Wrist and Cup

After introducing the vicon system and the egoSphere, it is important to introduce the two components which have the closest relationship to the use of egoSphere during the whole implementation process i.e. the hand wrist model and the cup.

In this project, the plan became to use the position relationships of the hand wrist model and the cup to predict whether the user is holding the cup or not. For example, if the hand wrist and the cup are far away from each other, then we could not say the user is not holding the cup with almost a hundred percent confidence. Afterward two modules were added to the egoSphere with the method placing the markers as mentioned in the previous chapter, and by using their objectIDs, which are the same across different databases, to stand for different objects.

A module named ObjectProvider in C++ was written to calculate whether the user is holding the cup and after adding the objects into the egoSphere, those values for Cartesian coordinates, radius, azimuth, elevation and rotation can be easily streamed out from the system (Fig. 15), thus the position as well as the movement of each object in the egoSphere can be presented and recorded clearly.

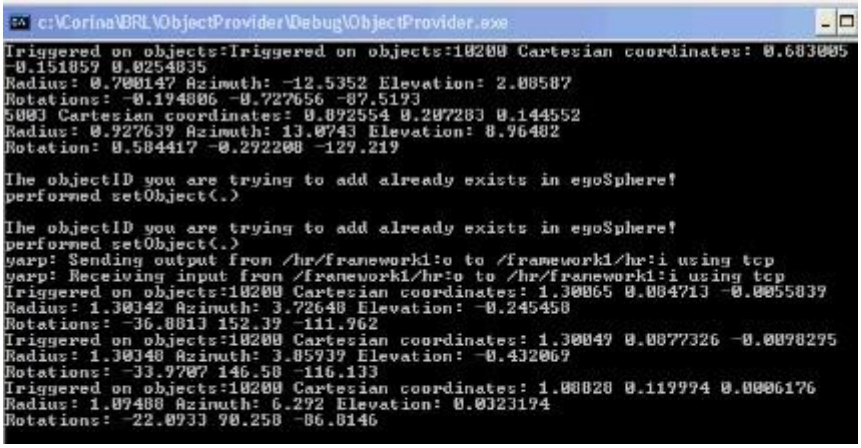



Fig. 15

The position relationships of the hand wrist and the cup that were taken into consideration here are essentially divided into two aspects: vertical distance and horizontal distance. It was decided that if the values for the two different distances are both within some sort of thresholds, then it can be said that the user is grabbing the cup. The thresholds for horizontal distance and vertical distance are certainly different, and were tested in the experiments by trying different values of these two thresholds to make them more accurate and more reasonable. As the user needs to wear the hand wrist model in this experiment, the horizontal distance is the distance difference from the center of the cup to the center of the hand wrist model as measured horizontally. Similarly, the vertical distance is that difference measured vertically.

However, the values of the positions may constantly be changing, and within a certain time period, the two distances within their respect thresholds for the first second may be detected, something known as “touching”, while for the following second the result of the detection may be “not touching”. These “touching” or “not touching” results come out once every interval. It is then that the problem can be solved successfully. The number of the result “touching” with a certain time period, which is actually the same time period we want the system to start checking whether the user is grabbing the cup from the start of the interval and report the result “touching” or “not touching” at the end of the interval, and if that number is larger than a certain value, then the result of the checking within that time period is “grabbing”, or just “touching”. That “certain value” here is also tested by the experiments for a more reasonable one, and should also be different for different safety requirements.

### 4.4.4 Robot Head

In the project, we have a robot head with two cameras placed at each side of it. Along with its controlling device, which is the software face lab here, this component is used to check the gaze of the user (Fig. 16).

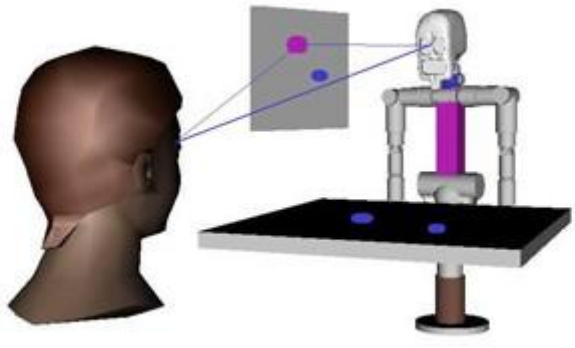


Fig. 16[33]

To use the software face lab as well as its cameras, the first step was to calibrate the position of the cameras. In addition, calibrating the user's gaze for the purpose that the information of the user's iris could then be remembered and logged by the device was needed. After that, the component could be used to check the user's gaze. Codes to stream data out from the system were written, and gazeObjectIntersection became a module for that use.

When the gaze of the user can be found and checked by the system, the data live is 1, otherwise it is 0 (Fig. 17). Intersection here means the user is looking at a certain object, and its value is just the egoSphere objectID of that object.

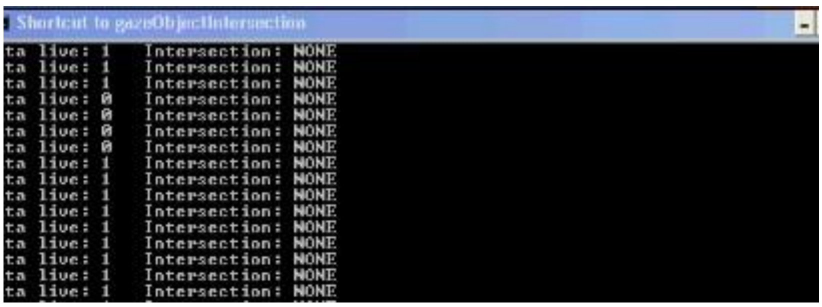



Fig. 17

As mentioned before, all objects in the egoSphere are assigned a unique ID, however, this egoSphere objectID for the same object may change from time to time. Although in the figure used for the vicon system, the egoSphere objectID is 0 for the robot head, here in the screen shot showed below (Fig. 18), the robot head is 1, and therefore it means the user is looking at the robot head. Fig shows the corresponding situation in a 3D way (Fig. 19), where the rectangle stands for the user's head, the large cubic item stands for the robot head, the small cubic item stands for the cup and the line between the rectangle and the large cubic item represents the user's gaze. Therefore, the scene shown in Fig also equates to the user looking at the robot head.

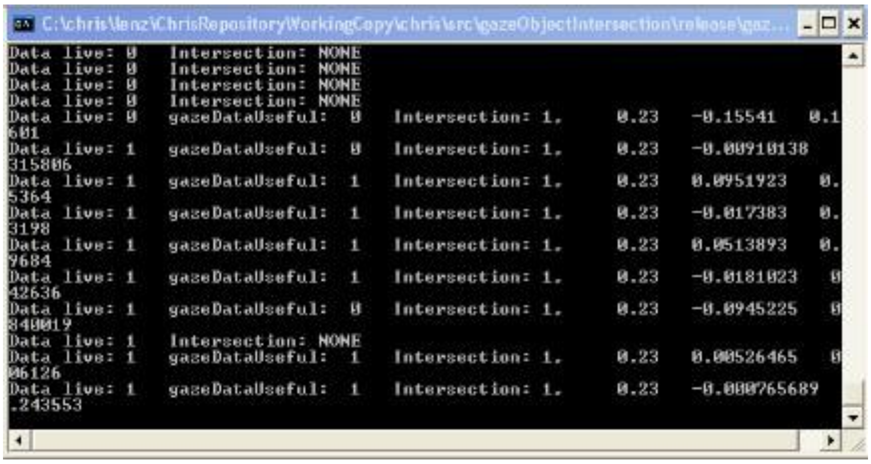



Fig. 18

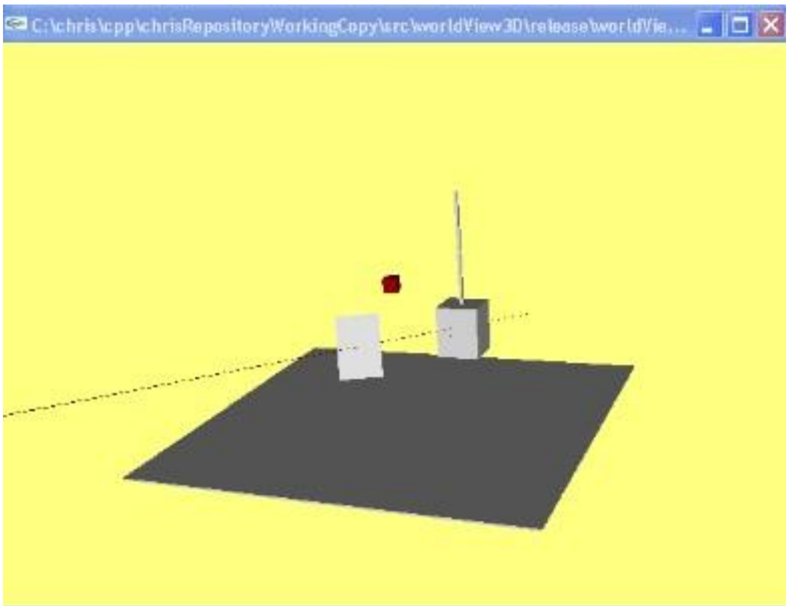


Fig. 19

After analyzing the meaning of the data, a conclusion can be drawn that if data live is 1 and gazeDataUseful is 1 plus the Intersection is the egoSphere objectID of the cup, then the user is looking at the cup. However, the same problem occurs in the way of calculating whether the user is holding the cup, and the calculation of whether the user is looking at the cup lasts for a certain time period as well. At first it was decided to settle this problem using the same means in the ObjectProvider module, and by writing the GazeCheck module to stream those essential data out from the gazeObjectIntersection. However, this fails for the same time period, which is three seconds in the project, the number of the results given out by the ObjectProvider module, “touching” or “not touching”, remains almost the same, and in fact is 20. Thus, there was only need to count the number of “touching” which was deemed adequate. However, at every three seconds, the number of frames given out by the

GazeCheck module continued to change. For the first three seconds, it may stream out 500 frames, and for the following three seconds, the number may be as large as 800. As a result, the number of “looking” could not be counted and creates the rule that if the number is greater than a certain value, then we can say the user is looking at the cup within the time interval.

In the end, a method to cope with this problem was devised, and the ratio instead of the number was used. It means, within the three seconds time interval, the number of frames was counted which means the user is looking at the cup, meanwhile the number of total frames was counted. In this program, the two numbers are named intersC and frameC separately. Then, the ratio is the result intersC accounts for frameC, in other words, intersC divides frameC. The following rule is the same with the ObjectProvider module, which is as if the ratio is larger than a certain value, then the user is looking at the cup. It is also true that this certain value needs to be tested again and again for a more reasonable and reliable result. In the screen shot below (Fig. 20), intersC and frameC are both 81, the rate, or the ratio, is then 100%, and the result after checking is surely that the user is looking at the cup.

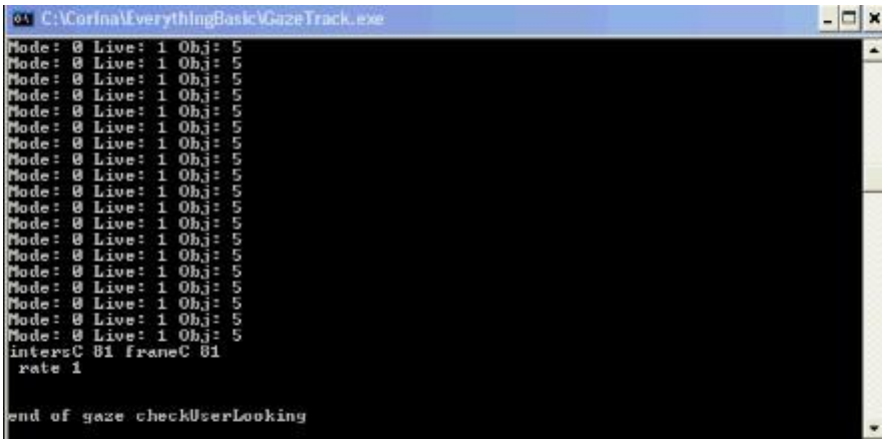



Fig. 20

## 4.4.5 Robot Hand

The robot hand in the project is used to move the cup (Fig. 21). It is a ssc-32 robot arm, which is a product of the company Lynxmotion located in the United States. The arm can move left and right, up and down; besides it has a grip which can be opened and closed. As a result, although this device is indeed quite simple, it is extremely suitable for the experiment and the project.

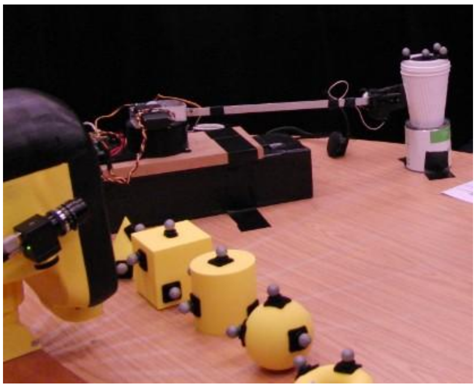


Fig. 21

The users' manual of the product was studied in order to understand the robot arm, and learn to control it with RIOS (Fig. 22), which stands for robotic arm interactive operating system. The extremely powerful program uses external inputs to affect the robot's motion for closed loop projects.


Fig. 22[34]

After gaining basic knowledge of the robot arm, a C++ code was written to control it, using Serial port. Finally, this was integrated with this part of code and into the whole Framework, thus control was gained from within the program. The initial position of the arm is 600 while the serve position is 900. This way to represent the height position of the robot arm is defined by RIOS itself, while these values are determined on our own and we pick up the values for "opening grip" and "closing grip" separately as well.

### 4.4.6 Voice System

The voice system plays an important role in the design of the whole project, and it is the bridge which links the user and the system. To be specific, the voice system explains the state of the system, gives out commands to the user, and takes in the orders and replies of the user.

In this project, RAD, which stands for rapid application development tool, has helped in the designing and implementation of the voice system. Actually, it charges most parts of the work. Although in the diagram, which is shown in the system overview section, the voice system links to the framework directly, there is a middle layer (Fig. 23), which is named YARP RPC. The output of the RAD is sent to this RPC, and then the RPC translates it and sends it to the framework.

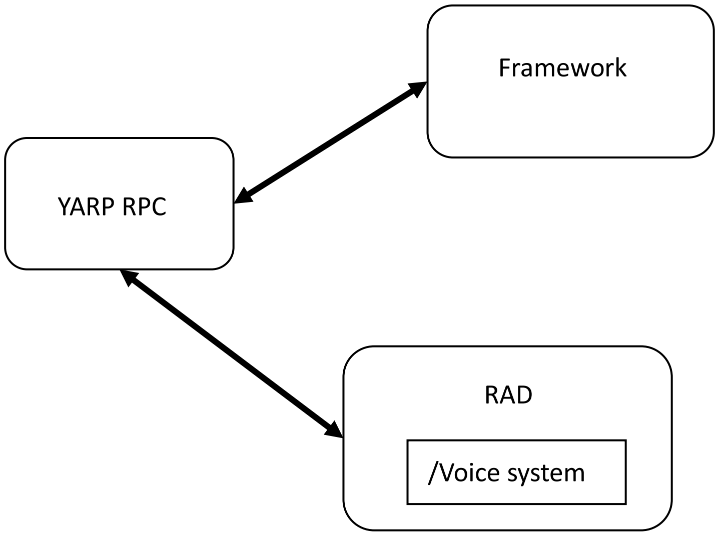



Fig. 23

As mentioned in the previous section, the basic and successful scenario for water choice is:
BERT2: Hello I'm Bert 2, do you feel like a drink?
USER: Yes.
BERT2: Water or Coffee?
USER: Water.
BERT2: Preparing the drink.
(The drink is prepared and moved to the serve position.)
BERT2: Please grab the cup.
(The user needs to look at the cup and grab the cub at the same time.)
BERT2: Enjoy.
(Grip is opened, and the user then can take over the drink.)
(System waits for 20 seconds.)
BERT2: Would you like another drink?
…
The basic and successful scenario for coffee choice is:

BERT2: Hello I'm Bert 2, do you feel like a drink?
USER: Yes.
BERT2: Water or Coffee?
USER: Coffee.
BERT2: Preparing the drink.
(The drink is prepared and moved to the serve position.)
BERT2: Please pay attention to the cup.
(The user needs to look at the cup.)
BERT2: Please grab the cup.
(The user needs to look at the cup and grab the cub at the same time.)
BERT2: Are you sure you have the cup?
USER: Yes.
BERT2: Enjoy.
(Grip is opened, and the user then can take over the drink.)
(System waits for 20 seconds.)
BERT2: Would you like another drink?
…

From the scenarios it is realized that most parts of the two scenarios are the same, and then they can be reused to make the system clearer and simpler (Fig. 24) (Fig. 25). After building the scenarios, items can be added to make it operate properly. The computer language which can be used to control the RAD voice system is TCL, which stands for Tool Command Language. Luckily this language does not need to be learnt in detail, and a double click on each icon in the chart and a modification of parts in the model is sufficient.

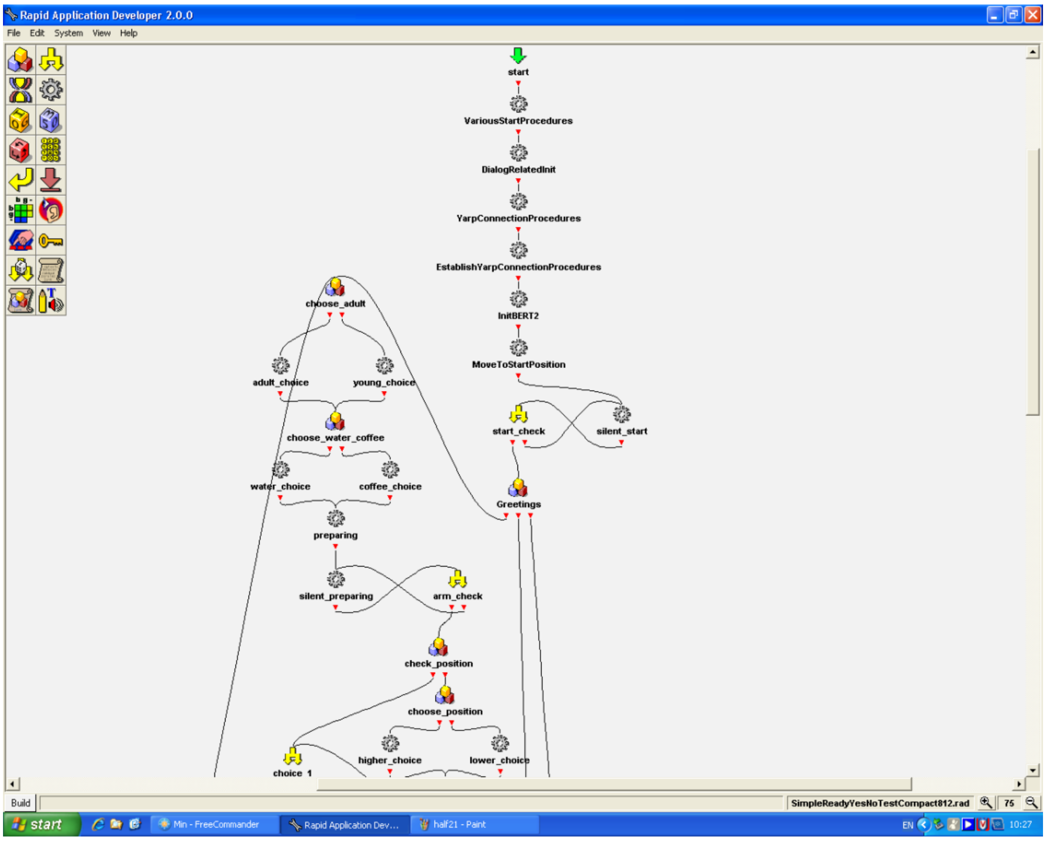

Rapid Application Developer 2.0.0
File Edit System View Help
start
VariousStartProcedures
DialogRelatedInit
YarpConnectionProcedures
EstablishYarpConnectionProcedures
InitBERT2
MoveToStartPosition
choose_adult
adult_choice
young_choice
choose_water_coffee
water_choice
coffee_choice
preparing
silent_preparing
arm_check
start_check
silent_start
Greetings
check_position
choose_position
higher_choice
lower_choice
choice_1
Build
SimpleReadyYesNoTestCompact812.rad
start
Min - FreeCommander
Rapid Application Dev...
half21 - Paint
10:27

Fig.24

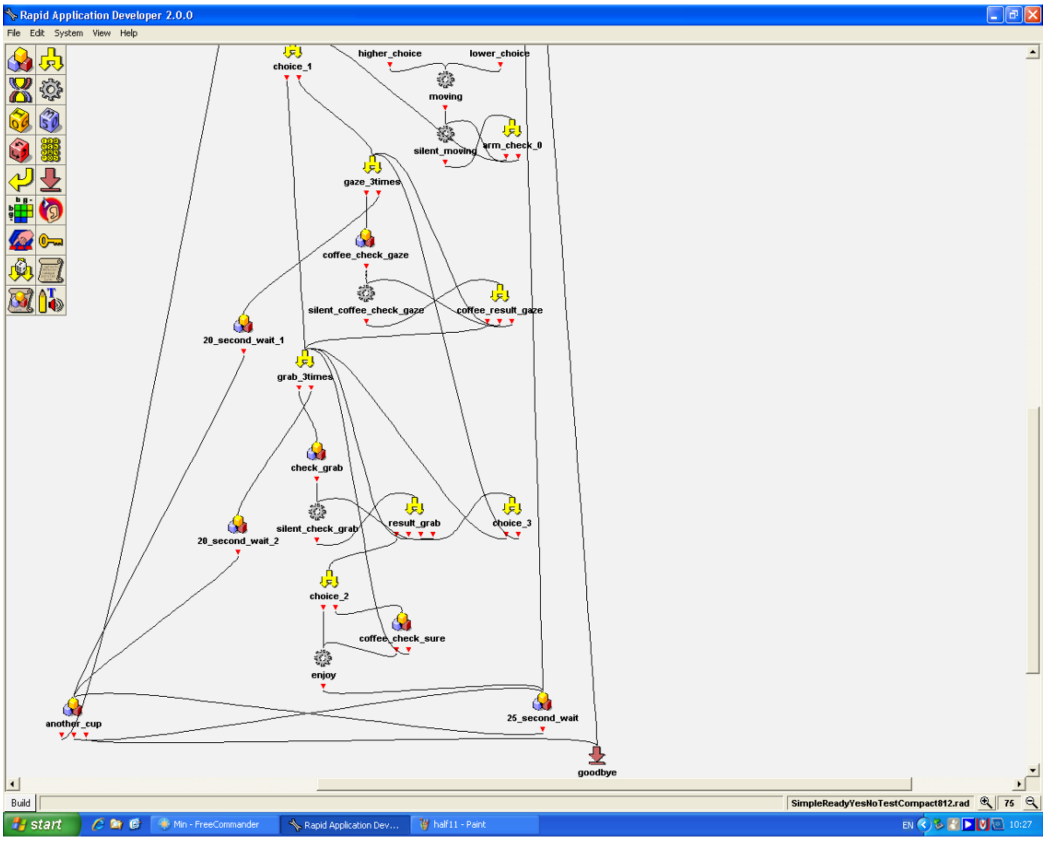

Rapid Application Developer 2.0.0
File Edit System View Help
choice_1
higher_choice
lower_choice
moving
silent_moving
arm_check_0
gaze_3times
coffee_check_gaze
silent_coffee_check_gaze
coffee_result_gaze
20_second_wait_1
grab_3times
check_grab
silent_check_grab
result_grab
choice_3
20_second_wait_2
choice_2
coffee_check_sure
enjoy
another_cup
25_second_wait
goodbye
Build
SimpleReadyYesNoTestCompact812.rad
start
Min - FreeCommander
Rapid Application Dev...
half11 - Paint
10:27

Fig. 25

Here, an example is presented (Fig. 26), and can be seen in the screen shot after double clicking on the icon “choose_water_coffee”. This is written: “Water or coffee?” under “TTS”, and when the system goes into this icon, it will say “Water or coffee” in voice. The items under “On Exit” is shown on Fig, which means when the system goes to the next icon, you will see the sentence “Here comes the Yarp RPC reply” plus a number, either 0 or 1, from another terminal which is named “RadPrinter”. Some sentences are printed which states the order to track the running status of the system. “After 200” means after the actions are listed, the system should wait for 2 seconds before going on to the following steps. Then, the tool is conveniently used to develop the voice system, to make the system talk in some given situations, to get information either from the voice of the user or from other systems, and to send out information to other systems as well.

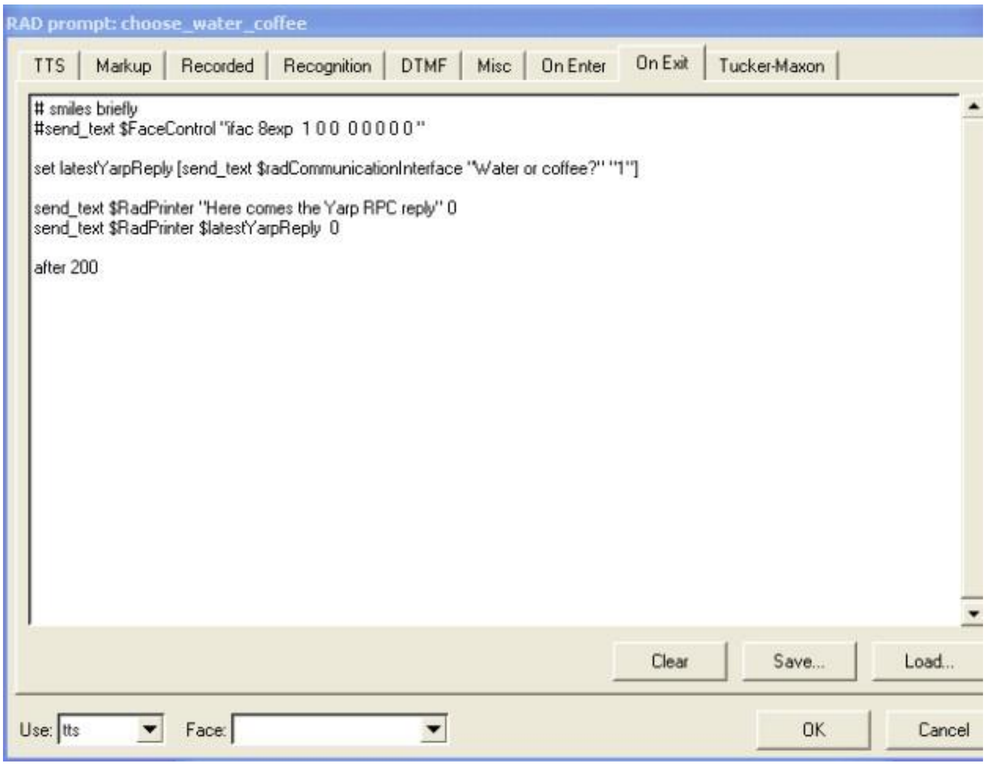



Fig. 26

After developing the voice system solely, the next step is to synchronize them with other modules. As the voice system can take in and send out information streams, we can control it in this way, in other words, when the voice system finishes one certain task and the following task of the project is due to finish in other modules, the voice system can send out data to the framework to trigger other modules to work; and vice versa.

### 4.4.7 Modules Summary

Up till now, we have given a brief illustration to some of the relevant but fundamental modules and devices. These components include YARP platform, egoSphere, hand wrist model, coffee cup, robot head, robot hand and the voice system. They can be independent from each other, as they can perform some certain tasks on their own; while in this project, they are synchronized and collaborate with each other to perform all the scenarios and the whole task as specified.

## 4.5 Summary

All in all, this chapter has demonstrated the design as well as the running process of the drink servant robot, and it also gives an overview of the whole system, whose interactions between each module are really complicated. Then it explains the importance and the use of each module, and illustrates the difficulties of synchronizing them as well.

The next section shows the work following this design and implementation phase, and experiments are carried out in the lab to test and validate this system. For the testing part, an essential and key point is the choice of several different rates/values for the measurement of experimental variables, for instance, the percentage of gaze-checking to classify user into “UserLooking” category. For the validating part, we need to see if the system is safe enough, live enough, or to just make the system as live as possible while not violating those safety requirements.

# Chapter 5: Experiment

## 5.1 Introduction

In this section, the aims of the project are firstly listed, which explains the rationale behind these experiments; and then the settings and processes of the experiment are also illustrated, in order to provide a detailed understanding of our experiments. Many people are invited to the lab to do the experiments, for a more accurate and reliable result.

## 5.2 Aims of the Experiment

In human-robot interaction, many detailed issues should be considered carefully to ensure the quality of that interaction, and that is particularly the matter in our context of a drink servant robot. Despite safety properties which are the cruxes, we should also guarantee quality of the service.

Hence, the aims of the experiments can be mainly divided into two parts. First, we need to see if the system is safe enough for different users while it runs continuously for many times; in addition, we also need to look for some special values which will play an important role in making the system meet the safety requirements as well as the liveness requirements, in other words, to find boundaries of those values between the safety and liveness properties.

The scenarios for both coffee and water have been introduced in an earlier section, and we can conclude from them that there are a set of values needed in order to test for the best one, which complies with both safety and liveness requirements.

According to the design, the interaction between the drink servant robot and the human for the first time turns up when the robot asks “do you feel like a drink”, and as a response the user should answer either “yes” or “no” or just “exit”. It involves the voice system module. Then, the number of times the voice repeats in the system should be taken into account. In other words, if the voice system cannot recognize the human’s answer, should it repeat the sentence “do you feel like a drink” continuously until an answer is successfully heard? In our opinion, if it constantly says those words while it not receiving a reply, it will drive the human mad. However, just as a coin has two sides, it also seems not especially lively if the robot says the sentence together just once. In that case if the human’s reply cannot be recognized by the system and even if the human does not give a reply at all, then the robot stops this service immediately and waits for some time before asking if the user would like another drink. Though a situation like this is acceptable, it may waste time and challenge the human’s patience, in other words, it violates the liveness requirements more or less. So, all in all, in our experiment we need to firstly figure out how many times the robot should repeat for each sentence until an answer is received.

After the water or coffee choice is made, the servant robot prepares the drink. In this project, this includes many steps such as the robot's hand moving to the initial position, the grip opening, the grip closing, and the hand moving to the serving position. From these steps, and together with the following steps, which involves the robot hand and the grip, we can clearly see that speed is the main concern, which it includes both the moving speed of the hand and the opening and closing speed of the grip. The speed problem is extremely vital as it is associated with "the safety of the cup" and the liveness requirements as well. For example, if the hand moves too quickly, the liquid in the cup may spill over; if the hand moves too slowly, the human may lose patience for the drink; if the grip closes too fast, the cup may have not been placed at the right position thus it may be knocked down; if the grip opens too fast, the human may not be ready to take the cup thus causing some unwanted trouble. Overall, in this experiment it was necessary to understand the values for these two different types of speed.

In addition, it is almost a general point of view that when dealing with machines, the response time plays an important role in the interaction. It should neither keep the human waiting for too long a time, nor react too fast and leave too little time for the human's response. Besides, it should comply with the design of the system too. For example, when the human has enjoyed a drink, how long should the system wait before asking "would you like another drink". Thus, all in all, in this experiment it was necessary to understand the different waiting/response time for different steps and different requirements in the system.

Furthermore, as described before, when conducting the gaze-checking work, it is important to check the gaze for a certain period, and during this certain period, if the ratio, which stands for the human is looking at the cup is above a certain value, it can be said that the user is looking at the cup. Thus, it is obvious that the certain period for gaze-checking and the certain value for user-looking should be experimented with.

Moreover, the way to check that "the human is grabbing the cup" relates to the vertical and horizontal distance from the center of the cup to the center of the human's hand. However it cannot be tested for only two values, which means it should not be only one value for the vertical distance and another for the horizontal one due to the different size of human palms. Although we are not able to determine it amply, for instance, to set several size ranges, we can roughly classify the palms into two categories, one for children, and the other for adults. Thus, overall, in the experiment the value of vertical distance and the value of horizontal distance for children and adults were required separately.

Finally, just as with the definition "the user is looking at the cup", the way to define "grabbing" is almost the same. Grabbing was also checked for a certain period, and when the number of grabs during that period is higher than a certain value, it can be said the human is grabbing the cup. In the design of the system, to produce a "grabbing" result the human needs to look at the cup while simultaneously grabbing it, then the system becomes synchronized and this "certain period" for checking gaze and checking grabbing should be exactly the same. Thus, all in all, we need to test the value for number of touching in the experiment as well.

In general, the six main issues that need to be experimented with are repeated for each sentence, the speed for moving the hand and opening/closing grip, the response and waiting time of the system, the period to check looking and grabbing, the ratio to define looking, and the number to define grabbing. In this experiment, different settings are given out to test these original values (we set some values according to our own small scaled test) and look for those values which are more accurate, reliable and acceptable.

## 5.3 Settings and Processes

### 5.3.1 Family Scenario

As introduced in the previous section, due to the different size of palms, we use different ranges of values for children and adults separately. However, excluding the values involved with palms, the other values are the same for all users, regardless of children or adults, females or males. In the program, we predefine these values so that any change to those values only happens at the very beginning of each piece of program, rather than needs to be modified for every place it appears.

However, the values cannot be combined to determine grabbing for children and adults, the scenarios need to be made a little bit more complex, which store the "role" as a variable (Fig. 27). This "role" has two values, and apparently one is "children" and the other is "adult". Besides, when the drink is at the given serve position, the new scenarios will ask "is the position okay for you to grab the cup" and move once higher or lower according to the human's answer. This plan is made based on the different height of all the members in a family. In the following steps, if the system detects the human is looking at the cup while not grabbing the cup, it may assume the position is not proper for the human to take the drink. The assumption is that the human is too tall or too short, or maybe the human's hand is broken and cannot stretch any further, so it needs to adjust the position in order to enable the human to grab the cup (Fig. 28). However, this decision to move higher or lower again here is made by the system itself, according to the information it has collected up untill now. For example, when the "role" of the human is "children" and the first movement is "lower", it is easy for the system to reach the assumption that the position is still too high for the human and it needs to move it even lower. Although it is not 100 percent confident, this kind of implication is always made in daily life and too some extent is reliable. Then, the system runs incessantly until a reply of "exit" is received.

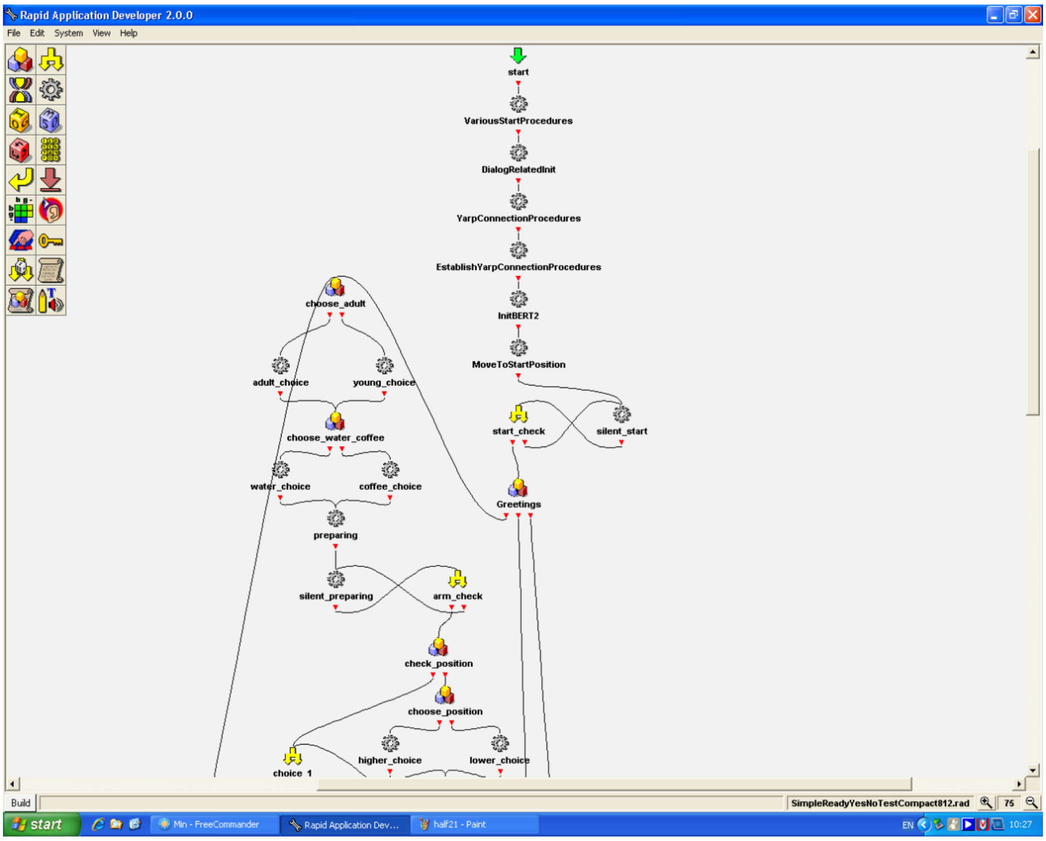



Fig. 27

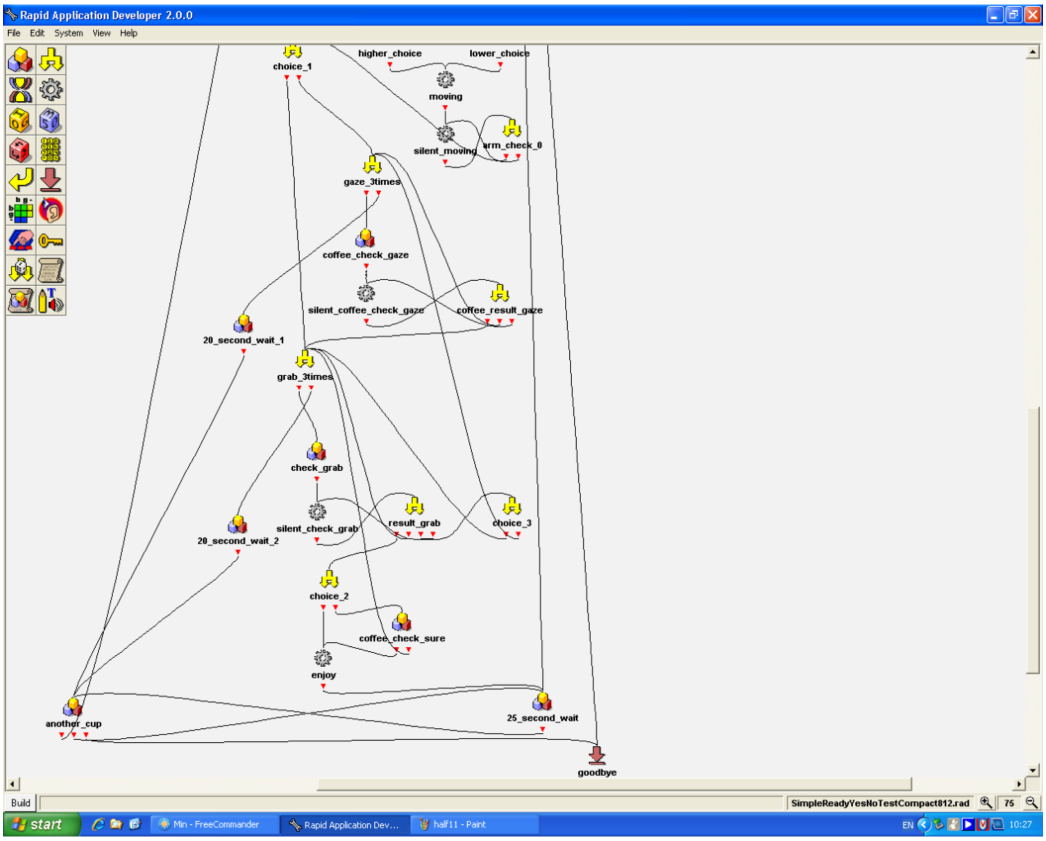



Fig. 28

## 5.3.2 Settings

Before starting the experiment, all values to be tested are given initial values. These default values are gained during the implementation work, and can be accepted. For example, the checking time for both gaze and grabbing is three seconds, which is neither too long nor too short. As mentioned before, too long a time may strain the human's patience, and too short a time may not be enough for the system's response as it needs time to read the data, process them and then produce a result followed by a reaction, so we had better find a compromise, and the "three seconds" was thought to be the boundary between the two aspects. Although it may be altered in the experiment, we use it as the initial value for gaze checking time and grab checking time.

To be more detailed, take the process of looking for initial ratio values for gaze-checking as an example (Table2). First, participants are asked to perform as required to, which are "continuously looking at the cup", "not looking at the cup at all", "looking at the cup intermediately" and "for the first half of three seconds, looking at the cup; and for the rest of the time not looking at the cup" separately. These actions are repeated for many times, and table2 is part of the data collected from a single participant. For example, when for the first time the participant is actually looking at the cup, the ratio calculated is 0.851 instead of 1, which in our mind is the represent of "absolutely looking". It is a similar situation when for the seconding time, the participant is in fact not looking at the cup but the ratio is 0.386 rather than 0. It is more complicated when the participant is "sometimes looking at the cup". A boundary between "looking" and "not looking" should be induced, which are apparently not 0 and 1. For different safety requirements, after many experiments, it is decided to make 0.75 as the ratio for "coffee looking", 0.25 as that for "water looking", while 0.4 is that ratio for those ask for a middle safety requirements between water and coffee.

Table2

| Action / times | Looking | NotLooking | InterLo | FirstLoThenNotLo |
|---|---|---|---|---|
| 1st | 0.851 | 0 | 0.45 | 0.2731 |
| 2nd | 0.787 | 0.386 | 0.343 | 0.343 |
| 3rd | 0.93 | 0.354 | 0.23 | 0.429 |
| 4th | 0.848 | 0.239 | 0.377 | 0.315 |
| 5th | 0.779 | 0.364 | 0.219 | 0.26699 |
| 6th | 0.867 | 0.201 | 0.429 | 0.363 |
| … | … | … | … | … |

To further verify these values and to see if they are safe enough and not totally violate liveness requirements at the same time, another experiment is carried out. Table3 shows the data collected from a single participant (Table3), where for the first 6 times the ratio is set as 0.75, and for the following 6 times it is set as 0.25, then for the rest times is 0.4. The participants can do as what they like to, either looking, or not looking, or looking intermediately, or other ways they may image. After the three seconds interval, they are required to tell that, in their mind whether their attention is focused on the cup. Compare the participants' answers with the outputs of the system, a judgment of whether the system is safe and live enough then can be reached. For example, for the first time, the ratio is set as 0.75, and although the participant think he is looking at the cup, the ratio calculated is only 0.56, which is surely less than 0.75 and is classified as "not looking", and this setting then can be recognized as safe enough; however, for the defined ratio 0.75, for many times even the participant himself think he is looking at the cup, the outputs of the system remains "not looking", in other words, the robot will not release the cup in any case and the user cannot take over the drink in any case, these situations are recognized as "too safe", and surely not comply well with liveness requirements.

Table3

| times | CalRatio | Output | RealAnswer |
|---|---|---|---|
| 1st | 0.56 | NO | YES |
| 2nd | 0.919 | YES | YES |
| 3rd | 0.323 | NO | YES |
| 4th | 0.177 | NO | NO |
| 5th | 0.255 | NO | NO |
| 6th | 0 | NO | YES |
| 7th | 0.595 | YES | YES |
| 8th | 0 | NO | YES |
| 9th | 0 | NO | NO |
| 10th | 0.169 | NO | NO |
| 11th | 0 | NO | NO |
| 12th | 0 | NO | NO |
| 13th | 0.083 | NO | YES |
| 14th | 0.305 | NO | YES |
| 15th | 0.81 | YES | YES |
| 16th | 0 | NO | YES |
| 17th | 0 | NO | NO |
| 18th | 0.656 | YES | YES |
| 19th | 0.268 | NO | YES |
| 20th | 0.344 | NO | YES |
| 21st | 0.386 | NO | YES |
| 22nd | 0 | NO | YES |
| 23rd | 0.047 | YES | NO |
| … | … | … | … |

Many other initial values are gotten using the same way, despite the vertical distance and the horizontal distance which are used to judge whether the human is holding the cup. Tools are used to measure the size of our palms, rather than just predict them according to our own perception; and then estimate the two different distance ranges based on these results.

Until now, there are two choices of roles: children or adults, two choices of drinks: water or coffee, two choices of movements: higher or lower, besides, it totally says “Please pay attention to the cup” for three times and another three times to say “Please grab the cup” as well. These choices lead to a variety of scenarios, and to produce desired results, carefully thinking was required about all the possibilities before building the experiment process and carry out the experiments.

### 5.3.3 Processes

Before the experiment, we sent out various invitation emails and invited many people to the lab to help us carry out the tests. These two forms are enclosed in the attachments of the emails to let the people know the instructions of the experiments and decide whether they would like to participate in the experiments.

Enough participants were gained, including children and adults, females and males, people who are familiar with knowledge of artificial intelligence, robots and computer science and people who are not. Besides, some of them had read the attachments of the email, while others had not, which meant besides children and adults, we are also able to classify the participants into naïve users and professional users.

In a real situation when a drink servant robot serves the human a drink the human may be doing some other things and his attention may not be totally focused on the cup and drink while the robot is handing over the drink, for instance, he might be writing his own stuff, he might be watching TV, or he might even be taking to someone else. Thus, he might only look at the cup once in a while, or look at the cup intermediately, or even not look at the cup at all! However, we still need to assure the safety of the service even in the worst case scenario. Therefore, we need to observe the safety and liveness properties as well as test the values in the experiments, and thus an attempt was made to talk to the participants while they were taking part in the experiment, trying to distract their attention in a normal way. After thinking out what needed to be part of the experiments, the experiments commenced.

All devices were readied, and place at the fixed location so as to make the situation more like what happens in a family in daily life and to gather precise result as well (Fig. 29).

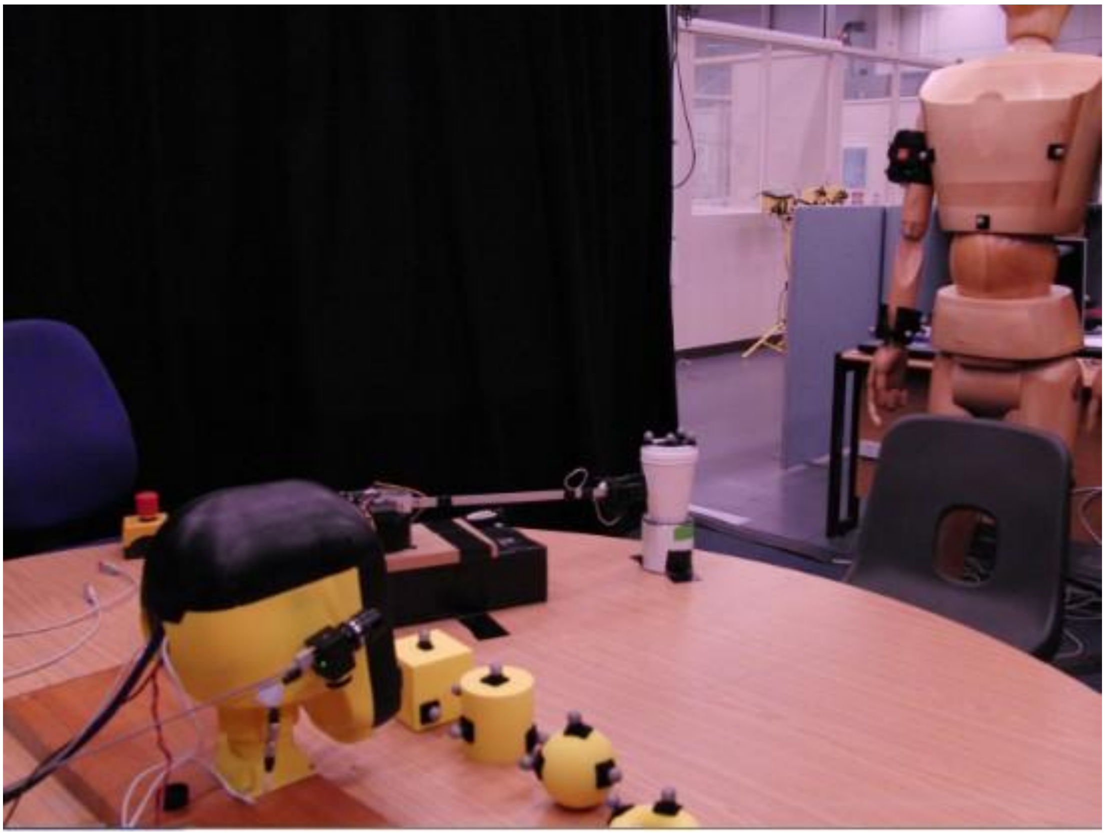


Fig. 29

To be specific, firstly, the robot head with its two cameras should be located, in the center of the room, which is also the center of the egoSphere, in order to make the gaze as well as other objects like the coffee cup and the hand wrist model better detected by the two cameras placed at each side of the robot head and by the eight cameras placed in the eight different positions of the roof. Besides this, the position of the robot hand was fixed as well as the coffee cup because the gaze-tracking system can only detect the gaze within no more than thirty degrees to its center. Moreover, due to the robot hand not being heavy and easily pulled if the human uses too much power, some rubber fabric was used to fix them in the right position, so as to make the experiments progress continuously. After all thus was carried out, the position of the chair was set and along with the Microphone everything was ready for the experiment.

As we have two scenarios, one is basic, and the other is the family scenario which includes the moving step after the original movement, we would like to test both of them in the experiment. To be specific, we use the basic scenario to look for and test the values as well as to verify the safety and liveness issues, and use the family scenario in order to see if it is safe enough while not violate the liveness requirements.

When the participants arrived, they were divided into two groups based on whether they had read the attachments of the email and understood the process before they came to the lab, which is to say, one group became professional users, and the other group naïve users. Several rounds were tested for a single participant.

The first to take part in the experiment was a boy, and before he started his gaze was calibrated. This step is designed for the system to remember the human's iris so that it can perform better when tracking the gaze. Then the participant's voice was calibrated, and this step is of the same use with the first one, just to make the system remember the participant's voice and to reduce the noise level when reading his replies or answers. Actually, the noise in the background affects the performance of the voice system quite significantly. Although when tested at home it works quite well, the voice system has some problems to understand the users' replies due to the noise at the lab which is also listed in the aims of the experiments, where we measure the palms' size of different participants so as to make the distance range more accurate.

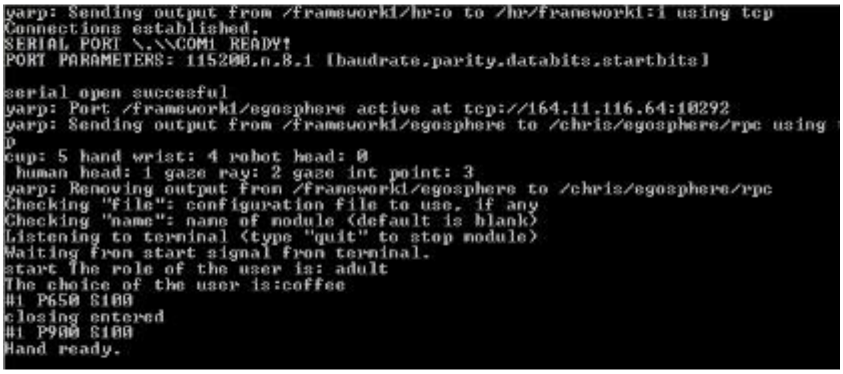
For the first two rounds, the boy's choice is water and then coffee, and the system does just what it is supposed to when everything is all right and when the user does everything correctly. In fact, the boy gives his whole attention to the robot and reacts as soon as the robot gives out commands and asks questions if it is not too fast. This causes a little trouble as the voice system is designed to take in the answer a short while after its own speeches finished, and we have to tell the boy to wait and then answer the question. However, except for this problem, everything works properly. According to our observation, it is safe enough for the boy.

For the third and the other rounds, just as designed we started to distract the boy's attention. As the system repeats "Please pay attention to the cup" and "Please grab the cup" three times each, we can talk to him or do some other things to achieve our goals, maybe after the system says "please grab the cup" for the first time, or after the second time, or even continuously, and the robot should not release the cup in any case if the user is not ready to take over the cup. From the experiments of the first boys, we can see it is at least safe enough.

However, it is not the same case for the liveness property. Through the interview with the boy after the experiments, it was found that sometimes even when the boy thinks his attention is mostly focused on the cup, the system still outputs the result "not looking" and asks the human to look at the cup. Also sometimes even when we see the boy is holding the cup, the system still detects the user is not grabbing the cup. Then it was realized that there might be some problems existing with the original values so adjustments were required. This was not considered disappointing as it is consistent with the parts of our intended aims. Thus, we changed each value several times to find one which performs well for both safety and liveness requirements.

The steps are almost the same for the other participants, the only things that were required to perform were calibrate the gaze and the voice, watch the performance, adjust values if needed, and then write down notes to record the results.

The figures below show the outputs of the framework module. The first two demonstrate the successful process where the role is an adult, he wants a drink of coffee, and would like the hand to be higher (Fig. 30) (Fig. 31). The other two demonstrate the process where the role is a young person, he wants a drink of coffee, and would like the hand to be lower (Fig. 32) (Fig. 33); for the first time the user is not looking, and for the second time the user is looking, then the system goes to check grabbing. As it is required to look at and grab the cup at the same time, this step actually checks the gaze again along with checks grabbing, and the checking result is “user not looking” and “user not grabbing”, thus it has totally checked gaze for three times and the robot hand moves to the initial position, waits for a few seconds and asks whether the user would like to have another drink.



```
yarp: Sending output from /framework1/hr:o to /hr/framework1:i using tcp
Connections established.
SERIAL PORT \.\\COM1 READY!
PORT PARAMETERS: 115200.n.8.1 [baudrate.parity.databits.startbits]

serial open succesful
yarp: Port /framework1/egosphere active at tcp://164.11.116.64:10292
yarp: Sending output from /framework1/egosphere to /chris/egosphere/rpc using
p
cup: 5 hand wrist: 4 robot head: 0
 human head: 1 gaze ray: 2 gaze int point: 3
yarp: Removing output from /framework1/egosphere to /chris/egosphere/rpc
Checking "file": configuration file to use, if any
Checking "name": name of module (default is blank)
Listening to terminal (type "quit" to stop module)
Waiting from start signal from terminal.
start The role of the user is: adult
The choice of the user is:coffee
#1 P650 S100
closing entered
#1 P900 S100
Hand ready.
```

Fig.30

```
closing entered
#1 P900 S100
Hand ready.
The user wants the hand to be: higher
#1 P1000 S100
before coffeePreProc()
SilentCoffeeGazeCheck
UserLooking

Looking
silentGrabCheck
startGaze

startHR

gaze result: UserLooking
Hr result: UserTouching

Grabbing
it's enjoy
```

Fig. 31

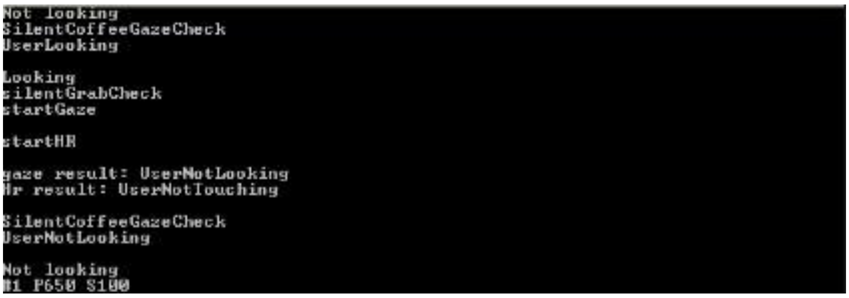
```
start The role of the user is: young
The choice of the user is:coffee
#1 P650 S100
closing entered
#1 P700 S100
Hand ready.
The user wants the hand to be: lower
#1 P800 S100
before coffeePreProc()
SilentCoffeeGazeCheck
UserNotLooking

Not looking
SilentCoffeeGazeCheck
UserLooking

Looking
silentGrabCheck
startGaze

startHR
```

Fig.32



```
Not looking
SilentCoffeeGazeCheck
UserLooking

Looking
silentGrabCheck
startGaze

startHR

gaze result: UserNotLooking
Hr result: UserNotTouching

SilentCoffeeGazeCheck
UserNotLooking

Not looking
#1 P650 S100
```

Fig. 33

Through these experiments, lots of data has been collected, where inductions can be made from to fulfill the requirements of the experiments, or in other words, find the boundaries/compromises between safety requirements and liveness requirements.

## 5.4 Results

After the experiments, a few rules and approximate values were found. Although they are not accurate due to the limitation of experiment times, it is believed that they can be treated as safe and as having liveness requirements and be built into human assisting robots in the future.

1. Check the gaze for a total of three seconds a single time, and decide within this period whether the user is looking at the cup;

2. Check the grabbing for a total of three seconds a single time, and decide within this period whether the user is grabbing the cup;

3. As coffee and water require different safety standards, where apparently coffee has a higher safety requirement, we define it if the user looks at the cup for more than or at least 75% of that 3 second period, then it can be said that the user is looking at the cup in the coffee scenario; and the value is 25% for that in water scenario;

4. As the different size of palms, the value of the distances used to decide whether

the user is holding the cup requires special attention. Here, for our system, we assign 0.05 meters as the horizontal distance for adults, and 0.02 meters as that for children; besides, 0.08 meters was used as the vertical distance from the center of the cup to the center of the palm for adults, and 0.05 meters is for children;

5. The threshold used to define whether the user is holding the cup should be different for coffee and water due to different safety requirements. We define this if within that three seconds period, the number of “touching” is larger than or equal to 8, then the human is holding the cup for the coffee scenario; and if that number is larger than or equal to 6, then the human is holding the cup for the water scenario;

6. To make it acceptable and equipped with liveness properties, every order and request of the robot should be repeated no more than three times, otherwise it seems a little bit ridiculous;

7. The moving speed of the hand should be 400 uS per second to make sure the drink will not spill over the cup;

8. The opening and closing speed of the grip should be 200 uS per second, to ensure the cup will not be knocked down and the cup will only be released when the user has taken the cup;

Many other similar rules have also been found, but overall, in human-robot interaction, especially in the context of human assisted robots, almost all the minor aspects should be taken into consideration to make the system and the process and the scenario safe enough, and after that live enough. It is hard to find the boundary between the liveness properties and safety properties, and if we want to achieve it, many more tests and experiments need to be carried out. Moreover, no single constraint can form an absolutely safe system in human-robot interaction, and rules should collaborate with each other to meet the human’s requirements for safety and liveness issues.

## 5.5 Summary

This section is mainly devoted to the processes and settings of our experiments. The number of participants can ensure the quality of the service, including both safety and liveness issues, and the various participants can ensure the safety in most of the daily cases, no matter for adults or for young people, no matter for females or for males, and no matter for naïve users or professional users.

The experiments are designed and carried out for the purpose of finding better values of different needs and requirements in addition to making sure the safety and liveness of the service quality, which is a more difficult than just ensuring the system is safe enough as the human's feelings towards the service is also of concern. The human should not be kept waiting too long and there should be enough time for the human to react while at the same time this kind of spare time should not be too long either. Information was collected through the recording and observation of the experiments, and also through interviews with the participants after the experiments.

After many experiments with different people, satisfactory results were gained. The approximate values were found, the safety and liveness properties of the system were tested and adjusted, and except for some small problems the experiment ran smoothly and these problems were solved by the end.

# Chapter 6: Conclusions

For a long time, Aimov's three laws of robot safety were recognized as the standards in much fiction, but on trying to apply Asimov's three laws to today's robots, there were immediate problems and the laws appear less applicable to today's robots. Hence, it is necessary to find a set of new laws or rules which are typically suitable to today's world.

The whole job of looking fornew rules may be extremely huge, so in this project, it is narrowed down to explore and identify such requirements for safe human robot interaction in the context of adaptable robot behavior, which is the behavior when a drink servant robot serves the drink to a human. During this process, the two issues of safety properties and liveness properties are carefully analyzed and investigated, and also potential safety reflexes are identified before implementing scenarios and validating the findings in those scenarios.

When it comes to the testing and validating phase, firstly some scenarios were selected and implemented, one for water and the other for coffee, as they required different safety standard; then the two scenarios were put into the context of a family, as different members of a family may ask different safety requirements. In the implementation phase, many different modules and components were used, which were provided by Bristol Robotics Laboratory, to fulfill our needs by synchronizing and simulating them to perform the whole task. Finally, the potential safety and liveness requirements were analyzed in those scenarios and the points which may affect safety and liveness requirements were understood and taken into consideration.

In order to test and verify those findings, many experiments were carried out in the lab. The boundaries between the values which were more satisfactory to safety requirements and the values which were more satisfactory to liveness requirements were found. After the experiments, many rules were discoverd, which in our mind could form the basis for a set of new laws for that kind of human assisting robots or for safe human-robot interaction.

During the period of carrying out this project, many challenges occurred and were conquered. Besides the study of new knowledge like YARP Platform and egoSphere, the control of many components like the mechanical robot hand and the voice system, and the synchronization of all the modules, "Robot" is nothing but a totally new area to us. Due to time limitation, a slight part of the objectives were not realized, including using formal properties or assertions to formally specify limits and constraints of some safety reflexes. But all in all, a set of rules have been found, which will be helpful to establish the standard of safe human-robot interaction in the future.

# Chapter 7: Future Work

## 7.1 Build in Learning Algorithms

In the steps of moving the hand to the serving position, it is typically easy in this project as the positions are mostly set and cannot be changed, thus it was only needed to assign the values which stood for those positions for the robot, and then the robot will move the hand to the destination. However, in real life, this may not be the case, for different users the serve positions are very likely to be different due to the different height and different body conditions of the humans. Then it was assumed that some learning algorithms could be built to meet these needs.

To be specific, when the robot serves the drink, it also collects information of the users, for example, the ages, the answers of moving higher or lower, and the final serve positions. Then, new user's information was also recorded, and classified into a category which is most similar to him, using algorithms like K-Nearest Neighbors or K-Means or some other algorithms like that. Finally, a value of serve position was assigned to him according to the serve positions of his neighbors in the same category.

Besides, many other algorithms can also be built into these scenarios and these kinds of human-robot interaction, just to make the robots more intelligent and more capable of helping humans, which is the basic goal of robotics study as well.

## 7.2 Number of Tests

In the project, experiments are carried out to test the values and verify the findings, but also in the project, the number of tests was not considered enough. To make the whole process even safer and further meet the liveness requirements, and also find better and more accurate boundary values, we think many more tests need to be carried out.

These tests again should include participants of different backgrounds, adult and young, males and females, professional users and naïve users, and even healthy users and disabled users. Besides, the scenarios of the tests should be more various, for example, the user may be watching TV while he is asking for a drink, or the user may be talking on a phone while he is asking for a drink, or the user may even be asked to do something else immediately after he asks for a drink.

A large number of tests are required in order to verify these findings, that is to say, if those findings and rules can guarantee the safety and liveness requirements in that kind of sundry and unique situations, it can be said with more confidence that they can form the basis of a set of new laws of human-robot interaction in the future.

# Bibliography


[1] "Visual Journal – Robot History". http://pages.cpsc.ucalgary.ca/~jaeger/visualMedia/robotHistory.html.

[2] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, "Robots in human environments: Basic autonomous capabilities," International Journal of Robotics Research, vol. 18, no. 7, 1999, pp. 684–696.

[3] "MegaGiant Robotics". http://robotics.megagiant.com/history.html

[4] Longman Advanced American Dictionary. Longman, 2nd ed., 2007.

[5] "The Free Dictionary". http://www.thefreedictionary.com/coexisting.

[6] M. Holliday, J. Tsai, "Industrial Robotics," WTEC Robotics Workshop. http://www.wtec.org/robotics/us_workshop/june22/.

[7] S.L. Anderson, "Asimov's 'Three Laws of Robotics' and Machine Metaethics," AI and Society, vol. 22, no. 4, 2008, pp. 477-493.

[8] A. Sloman, "Why Asimov's Three Laws of Robotics are Unethical," 27 July 2006; www.cs.bham.ac.uk/research/projects/cogaff/misc/asimov-three-laws.html.

[9] C. Allen, W. Wallach, and I. Smit, "Why Machine Ethics?" IEEE Intelligent Systems, vol. 21, no. 4, 2006, pp. 12-17.

[10] M. Moran, "Three Laws of Robotics and Surgery," J. Endourology, vol. 22, no. 8, 2008, pp. 1557-1560.

[11] R. Clarke, "Asimov's Laws of Robotics: Implications for Information Technology Part I," Computer, vol. 26, no. 12, 1993, pp. 53-61.

[12] R. Clarke, "Asimov's Laws of Robotics: Implications for Information Technology Part 2," Computer, vol. 27, no. 1, 1994, pp. 57-66.

[13] J.M. Bradshaw et al., "Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction," Agents and Computational Autonomy: Potential, Risks, and Solutions, M. Nickles, M. Rovatsos, and G. Weiss, eds., LNCS 2969, Springer, 2004, pp. 17-39.

[14] Robin R. Murphy and David D. Woods, "Beyond Asimov: The Three Laws of Responsible Robotics," IEEE Intelligent Systems, vol. 24, no. 4, 2009, pp. 14-20.

[15] "Trust me, I'm a robot". http://www.economist.com/displaystory.cfm?story_id=7001829.

[16] Y. H. Weng, C. H. Chen and C. T. Sun, "Safety Intelligence and Legal Machine Language: Do We Need the Three Laws of Robotics?" National Chiao Tung University, Taiwan, 2007.

[17] "Robot Safety," Industrial Welfare Division, Department of Labour, Private Bag, Wellington, New Zealand, 1987.

[18] Susan Owicki and Leslie Lamport, "Proving Liveness Properties of Concurrent Programs," ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, pp. 455-495.

[19] James H. Graham, "Research Issues in Robot Safety," Proceedings of the First International Conference on Ergonomics of Hybrid Automated Systems I, 1988, pp. 477-482.

[20] R. D. Kilmer, "Safety Sensor Systems for Industrial Robots," SME Robots 6 Conf., March 1982, pp. 479-491.

[21] Purves. Neuroscience: Third Edition. Massachusetts, Sinauer Associates, Inc, 2004.
[22] Fusco F., Gallerini R., European Robotic Arm: the Problem of Preventing Collisions, 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation, "ASTRA 2000", 5-7, December 2000, ESTEC, Noordwijk, The Netherlands.
[23] Noriyuki Tejima and Dimitar Stefanov, "Fail-Safe Components for Rehabilitation Robots – A Reflex Mechanism and Fail – Safe Force Sensor," Proceedings of the 2005 IEEE, 9th International Conference on Rehabilitation Robotics, June 28 – July 1, 200, Chicago, IL, USA, pp. 456-460
[24] European Norm EN 775: Safety of manipulating robots, Berlin, Beuth-Verlag, 1996.
[25] ISO 10218-1: 2006(E): Robots for industrial environments – Safety requirements – Part 1: Robot, August 2006.
[26]http://www.used-robots.com/robot-education.php?page=industrial+robot+safet y.
[27] http://www.sis.se/popup/iso/isotc184sc2/about_work_background.asp.
[28] ISO/IEC Guide 51: 1999, Safety aspects-Guidelines for their inclusion in standards.
[29] ISO 12100: 2003, Safety of machinery-Basic concepts, general principles for design.
[30] ISO 14121: 1999, Safety of machinery-Principles of risk assessment.
[31] http://www.brl.ac.uk/projects/chris/index.html
[32] Stephane Lallee, Severin Lemaignan, Alexander Lenz, et.al "Toward a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception".
[33] http://www.brl.ac.uk/projects/chris/index.html
[34] http://www.lynxmotion.com/p-419-rios-ssc-32-arm-control-software.aspx.