DEPARTMENT OF COMPUTER SCIENCE

I Robot, I Think

Elena Corina Grigore

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering

May 2012 | CSMENG-12

Declaration

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Elena Corina Grigore, May 2012

Executive Summary

Human-assistive robots can become an important part of our future if we manage to design useful and safe machines that will adequately interact with humans in helping them to achieve various tasks, examples are robot waiters, personal care robots helping patients in hospitals during recovery. In this project, my focus is to explore how human-assistive robots can acquire useful behavioural adaptations while remaining within safe boundaries. Particularly, the investigation focuses on ensuring that a drink is safely handed over to a human, taking into account how a human user reacts to realistic variations in the environment. The entire project was performed at the Bristol Robotics Laboratory (BRL), using BERT2, an upper-body humanoid robot, equipped with seven degrees-of-freedom (DOF) for each arm [1].

The overall aim of my project is to make sure that the robot can hand a drink to a user in a safe and predictable manner. I achieved this by building a model of the interactions between the robot and the human. This model is used by the robot to assess the situation at any stage during the interaction and estimate its state at any moment in time. I then added an extra layer which models the user's intentions and reactions to changes in the environment to aid in the decision making process. This reflects the "theory of mind" concept, discussed in more detail in the following sections. The two resulting systems, the one based on the initial model and the extended one which includes modelling the user's intentions, were then compared in order to derive what the implications are on the safety aspect of this close human-robot interaction scenario.

To summarise, the main contributions and achievements in my project are as follows:

- I initially started from an implementation of a basic control loop for the handover of a drink to a human on the BERT2 robot at the Bristol Robotics Laboratory as described in my recent TAROS paper: http://www.cs.bris.ac.uk/Publications/Papers/2001403.pdf.
- I implemented a Hidden Markov Model in MATLAB in order to create the basic model of the system, which allows the robot to estimate the system's state of interaction and, based on this, take the decision of releasing the cup to the user. I implemented two versions of the model (780 lines), in order to reliably estimate the state of interaction. I implemented the decoder for this state (Viterbi Algorithm) in C++ as a module which communicates with the other modular components of the system and controls the robot (620 lines). See subsection 4.2.1.
- I spent 50 hours investigating the Psychology aspects which uphold the layer modelling the human's intentions, both by working with an experimental psychologist and by reading relevant papers.
- I implemented the extended system in C++, in 590 lines. See subsection 4.2.2.
- I implemented two extra modules in C++, in order to control the robot: a human gaze estimator which is able to inform the system where the user is looking (430 lines), and a module which tells the system exactly when the user is touching the cup (340 lines). See subsection 4.2.2.
- I performed experiments using both systems and compared the results. See subsection 4.3.

Acknowledgements

I would like to deeply thank my supervisor, Dr. Kerstin Eder, for her continuous help and guidance throughout this project. This thesis would not had been possible without her support and suggestions. Dr. Ute Leonards has greatly helped me to unlock the psychology element for my project, and so I would like to express my gratitude.

Another thank you goes to Dr. Alex Lenz and Dr. Sergey Skachek from the Bristol Robotics Laboratory who have helped with the workings of the BERT2 robotic platform. Their help was truly invaluable. Prof. Tony Pipe and Prof. Chris Melhuish have been my supervisors at the BRL and I would especially like to thank them for their support and guidance.

Table of Contents

1 Context and Motivation

1.1 Project Topic

The "I Robot, I Think" project aims at modelling a system which can be used in the close human-robot interaction scenario of the robot serving a drink to a human user. The primary concern of this handover is making the action a safe, yet useful one. The process through which I achieved this followed the following steps:

1. I started with an implementation for a handover of a drink to a human on the BERT2 robot, which is described in my recent TAROS paper and which focused on setting up an environment to investigate safety and liveness aspects in the context of human-robot interaction [2]. The analysis puts an emphasis on safety and liveness as being key aspects of the trustworthiness of a robot, which is necessary in order for the user to be comfortable with interacting with the robot. The safety aspect is an essential one, yet an easy to understand requirement when considering a robotic assistant. In order for the robot to be able to engage in close interaction with a human, its developers need to ensure it is safe to so. This means the user will have a guarantee that he or she will not be harmed in any way by the robot. Liveness, however, means that the robot will eventually do something useful. This is in contradiction with the safety aspect, since the more dynamic and adaptive the robot will be, the more it will satisfy liveness constraints, i.e. it will be doing something very useful for the user, but the harder it will be to prove or even determine its safety. Thus, an equilibrium needs to be found between safety and liveness.

During an earlier summer project, I indentified the safety and liveness properties required by a hot-drink serving robot scenario and performed experiments to investigate if they indeed hold. This initial set-up gave me an insight into building a more complex model of the interaction to allow for the robot to take actions by taking into account what the user is most likely to do in a successful handover scenario.

2. I created a model of the system by using a Hidden Markov Model (HMM). This helps the system estimate the state the interaction is at any moment in time by looking at the motor current values from the robot's arm and hand. These values change accordingly when the robot is grabbing the object, moving with it, or when the user is grabbing the cup. Based on this estimation, the robot can then take the decision of releasing the cup, i.e. if the interaction is estimated to be in the state of the user grabbing the cup, the robot releases the cup.

The HMM model also assigns a level of confidence in the state by computing the probability of the system being in that specific state at that moment in time. Whenever the system estimates the state as being "user is grabbing the cup", if the confidence level is high enough, the robot will think it is safe to release the cup. The computation of the confidence level mimics the expectations a user has when trying to take an object from someone else. If the user does not feel the force of the other part weakening almost immediately, he or she will take his hand away quickly. Thus, the probability of the system remaining in the "user is grabbing the cup" state decreases as time goes by, in order to reflect this. Although it might seem counterintuitive in the first instance, this actually tries to model the human-human interaction for an object handover. However, the fact that the system does not have much time in order to assess the safety of releasing the cup in such a quick manner indicates that further reassurance would be needed to take the decision. As such, I extended the system by adding an extra layer, as explained further.

3. I extended this system by adding a layer on top of the existing model. This layer models the user's intentions and future reactions to changes in the environment. This represents a subset of beliefs, goals, and desires, which are attributed by one person to another during their interaction. The complete set is referred to as a "theory of mind" and, because it allows humans to understand the actions and expressions of others within an interaction, it is believed to be an important concept for robots to be able to interact as naturally as possible with humans [3].

In order to find a suitable implementation for this layer, I turned to the experimental psychology field for inspiration and collaborated with Dr. Ute Leonards, whose main research interests include visual perception and visual attention, joint attention, and eye movements [4]. The collaboration and further research into this area helped me formalize the essential actions a person does when engaging in a handover with a robot. One of the most important elements for being able to tell if a human's attention is focused on the object of the handover is visual attention. The human will always look, even if briefly or not continuously, at the object, before he touches it. Thus, the basic sequence of actions can be defined as: human is not looking at the object (he or she is looking at the robot perhaps or wandering around with the eyes), human looks at the object, human looks away (usually at the robot's head), human touches object in order to complete the handover. The robot will only release when this sequence is completed, thus interpreting this sequence as the user's intention of grabbing the cup. These findings are discussed in detail in subsection 4.2.2.

4. I performed experiments with subjects using both systems. This allowed me to compare the two systems from a safety viewpoint, in order to assess if adding the "theory of mind" does indeed impact on the outcome of the handover, i.e. if it reduces the number of times the cup is dropped or, more generally, the number of times the handover is unsuccessful. The comparison proves that the representation the robot has of the system is important and that it affects the decision making process. This, in turn, affects the safety level of the system, as well as how safety can be proven.

1.2 Topic Importance

The field of Robotics has many applications in the real world which have either already proved their benefit, or have a huge potential which needs to be unlocked. Industrial robots have long been used in various industries for performing tasks more efficiently than humans and in an automated way. The world of personal robots, however, is an emerging industry which needs to deal with complex and very important issues before it can offer its benefits to its users. These problems include achieving an adequate level of safety, proper human-robot interaction, useful performance, and affordable costs [5].

The importance of exploring what the necessary steps are for ensuring the technology will be able to make the transition from research into industry was acknowledged in September, 2010, at the joint EPSRC and AHRC Robotics Retreat, which discussed robotics and its applications in the real world, as well as the benefits it promises to offer society [6]. One of the principles formulated at the workshop was: "Robots are products. They should be designed using processes which assure their safety and security" [6]. The only way that personal robots will manage to get into industry is by assuring they will not harm humans. They are, of course, pieces of technology which should be protected, from both the seller's perspective and from the owner's. However, the most important element is making sure that robots can be trusted and that humans are comfortable using them, otherwise robotic assistant will remain just a vision.

It is clear that rules, standards, and tests which ensure safety should be adopted by the robotics industry in order for such products to be able to be marketed. In the world of industrial robots, safety is generally guaranteed by physical boundaries - powerful, industrial robots are isolated from human workers and there is no direct contact with them. Personal robots, however, depend on close human-robot interaction scenarios in order to achieve their tasks. As such, safety regimes within this area need to be thoroughly investigated to become on par with safety levels which are put in place in a multitude of consumer markets: industry kitemarks, British and international standards, testing methodologies for software to ensure they are bug-free, etc. [6]. Another useful example to look at is the aircraft industry, which has been dealing with safety critical systems for decades. Aviation software products must comply with stringent rules and regulatory guidance in order to pass certification [7].

Useful robots will have behavioural adaptations and will dynamically take decisions depending on the changes in the environment in order to be useful in a variety of situations. Even robots which do not learn during a certain task should be intrinsically safe - this means they should incorporate safety at all levels: mechanical, control, software, human interaction. Although there has been work done with respect to lower-level safety of robots, restricting movements at the mechanical level, the high-level behaviours of assistant-robots during interactions with people have not yet been handled. This topic is of extreme importance if a robot is to pass certification in order to be marketed and trusted to interact with human users.

The burden of proving the safety of personal robots will fall with the developers of such products, be it the researchers or engineers who create the software or the company which sells them. The benefits of discovering a reliable way of certifying a robot's actions as safe are going to impact both research and industry.

Certification is a process of negotiation with a certification authority, which sets up safety standards. The interested party (developer, researcher) then needs to demonstrate that the product indeed follows these standards, which can be achieved through verification and validation. The former is the process of confirming that a system matches its formal requirements, while the latter is the process of confirming that the system results in the intended behaviour once the system is embedded in its target environment [27]. Techniques range from formal safety verification, through testing, but a method for thoroughly and precisely proving the safety of a robotic assistant is an open ended research question which is currently only starting to be properly investigated. New techniques which take into account the robot's high-level mechanisms of taking decisions during a process can prove to be invaluable within this area. Thus, studying and analysis what effects the "theory of mind" concept has on the safety issue might reveal the necessity of new verification and/or validation methods.

Personal robots can be of great use in a myriad of situations, be it for mere interaction with humans or for helping patients during recovery in a hospital. Robots which are meant to help people can be classified by using the following categories, as described in [8]: Assistive Robotics, Socially Interactive Robotics, Socially Assistive Robotics. These categories are in fact not mutually exclusive, as a robot's tasks can have a high degree of complexity. Assistive Robotics (AR) is generally viewed as robots assisting people with physical disabilities through physical interaction. It includes rehabilitation robots [9], [10], [11], [12], [13], wheelchair robots and various mobility aides [14], [15], [16], [17], companion robots [18], [19], [20], manipulator arms for the physically disabled [21], [22], [23], and educational robots [24]. The environments of use of such robots include schools, hospitals and homes [8]. According to [8], the term Socially Interactive Robotics (SIR) was first used by Fong [25], to describe robots whose main task was some form of interaction, in order to differentiate between social interaction and teleoperation in HRI. The goal of the robot is to develop a close interaction with the user. Finally, Social Assistive Robotics (SAR) is the intersection of AR and SIR, with a goal of assisting human users not through physical contact, but through social interaction [8]. SAR is also defined as the "pursuit of creating robots capable of exhibiting natural-appearing social qualities" [26]. SAR's main application domains include care of the elderly, care of individuals with physical recovery/rehabilitation and training needs, and care of individuals with cognitive and social disabilities.

The hot-drink serving robot scenario interacts with users on two levels. First it requires direct, close physical contact, meaning the findings which imply safety aspects are highly relevant to all applications that fall under the SAR category. Second, the user's intentions are modelled and the user benefits from social interaction with the robot. It is true that the interaction's purpose is not to help the user with rehabilitation, however, the high-level concepts are the same and the same verbal and non-verbal elements of communication are employed in order to keep the user engaged and predict his or her intentions as would be the case in a rehabilitation situation. This makes the scenario relevant for the SIR domain as well as for SAR.

1.3 Challenge and Significance

Technology has reached a point which makes the Asimovian dream come a bit closer to becoming reality. Scientists and engineers are capable of building complex robots capable of intricate movements and suitable for a multitude of different tasks. This is possible due to advances in the technical field, as well as in the algorithms used to control these robots. Various learning methods exist which make for dynamic and adaptive systems, capable of dealing with diverse scenarios and of helping people in a myriad of situations, from automating dull tasks so that humans can focus on the challenging ones, through helping patients during recovery in a hospital, to interacting with the elderly. No matter what the benefit is, the more complex the scenario, the more adaptive the robot needs to be in order to be useful. This creates the challenging issues of demonstrating safety, reliability and trustworthiness of such personal robots. Formal standards need to be created by industry in order to ensure that when robots will enter widespread use, no harm will come to the humans interacting with them. As a consequence, reliable methods of demonstrating safety need to be developed and perfected. This is a novel, open research question, which can benefit from being studied from more than one perspective.

By integrating knowledge about the process model into the system, the current project aims to tackle the safety issue from two directions. First, the implementation of a learning algorithm which assigns a confidence level to the state estimation makes the system take the choice of releasing the cup only when it thinks it is safe to do so. This is treated as a basic level of satisfying the safety property which refers to the information available directly from the robot's arm and hand which holds the cup and interacts with the user. Second, the integration of a model of the user's intentions and reactions takes into account the higher-level safety conditions which should be satisfied in order for the robot to release the drink. It adds information about what the expected actions of a user in a handover situation are and makes the robot react more safely when this sequence is not followed, i.e. the robot does not release the cup and either puts it back or takes further steps in order to ensure the user still wants the drink.

The integration of a "theory of mind" in the project and the findings that the process through which the robot takes a decision is indeed better and proves to be safer than the system which does not include this layer is of relevance to any HRI scenario which involves social interaction. It is also a building block of the even more challenging task of creating and implementing nested levels of the "theory of mind", which would make the robot capable of representing what it thinks the human's beliefs, goals and desires are, what it thinks the human thinks about its own set, and so on. It is believed that social interaction requires a focus on the issues theory of mind research addresses [3]. The "theory of mind" framework is also used in [28: 311], but from the perspective of the user, researchers applying a "mindreading mechanism" that is indispensible in human-human interactions to a model of human-robot communications. They implemented a robot interface system that applied the proposed model and carried out experiments which upheld the hypothesis that a human can estimate a robot's intention with ease by reading the robot's mind, as well as understanding the robot's unclear utterances made by synthesized speech sounds.

Haptic interaction in human-to-human handovers was investigated and characterized in [29: 9-10]. However, this is one of the few studies which focused on characterizing the complete interaction and it only addresses the grip and load forces and their dynamic. Human motion when reaching for an object (before the actual handover) has been studied by many researchers, and for various modes of handover, but most research focuses on human-robot handovers or on single-person tasks of handling objects. The studies include the kinematics of the motion leading to the handover for human-robot handovers. However, in my project, the model of the user's actions needs to take into account the approximate timing for each action. This way, if the time between when a user last looked at the object before the handover and when he or she actually touches the cup is too large, the model warns the system that it is unsafe to release the cup. A lack of literature in the timings of the actions a person does before the handover and, especially a lack of literature involving visual cues representing the human's focus of attention in such an interation, makes this project step into novel territory.

The above discussion highlights the importance of the presented issues within the social robotics field and is a step taken towards creating safer and more trustworthy personal robots which interact with humans in close HRI contexts.

1.4 Project Main Aims and Objectives

The overall aim of my project is to make sure that a robot can hand a drink to a human in a safe and predictable manner. This is achieved by building a model of the interactions between the robot and the human. This model is used by the robot to assess the situation at any stage during the interaction and to make appropriate decisions during the hand over. Particular focus of my work will be to ensure that the robot is pro-active and guides the user towards the goal when the handover is not successful, but that it also remains useful by denying the drink to the user when it considers it not to be safe to release and asking the user if he or she wants a new drink.

To reach this aim, the specific objectives of my project are:

1. Survey literature from the robotics field to form a solid background, focusing on safety concerns and models of HRI which affect how a robot gains knowledge of its environment and takes its decisions.

2. Implement a robust mechanism for the robot to estimate the state the interaction is in with respect to the cup to be handed over (e.g. the robot is holding the cup, the user is trying to get the cup from the robot, etc.) with a certain level of certainty by assigning probability values to states.

3. Understand the "Theory of mind" - performing research into how to model the user's intentions within the field of experimental psychology. Implement a model which takes into account what the user is expected to do in a successful handover and integrate this knowledge in the decision process.

4. Extend the system from 1. to reflect the human’s interactions and potential reactions to the handover scenarios by transferring the knowledge gained in step 2. This is achieved by building a new layer modeling the user's intentions above the layer that represents the robot’s states.

5. Investigate if the extended system (from 4) is safer than the original one (from 2), including investigating the effects of the "Theory of mind".

2 Related Work and Project Technical Basis

2.1 Related Work

"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." [30] Asimov's stories sparked people's imagination, but they also proved how difficult they were to apply in complex real-world situations and that subtle interpretations of certain types of behaviour ("right", "wrong", "helpful", "harmful", etc.) can lead the robot to take an unexpected or incorrect decision. Asimov's laws are based on functional morality and assume that robots have the ability of making moral decisions by having sufficient agency and cognition, but when critiqued in [30], the analysis determines the authors to put forth a set of alternative laws which sum up the critical research questions facing robotics today.

	Asimov's Laws	Alternative Laws
1	A robot may not injure a human being or, through inaction, allow a human being to come to harm.	A human may not deploy a robot without the human-robot work system meeting the highest legal and professional standards of safety and ethics.
2	A robot must obey orders given to it by human beings, except where such orders would conflict with the first law.	A robot must respond to humans as appropriate for their roles.
3	A robot must protect its own existence as long as such protection does not conflict with the first or second law.	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.

The two capabilities believed in [30] to be essential for robots to have are responsiveness and smooth transfer of control. When analysing the proposed alternative laws, it can be observed that they remind researchers and developers of their legal and professional responsibilities.

One of the most important flaws of Asimov's first law is that it considers safety from the perspective of the robot, making it responsible for the entirety of its human-robot interactions and it forgoes serious practical, theoretical, and legal limitations [31], [32]. From a legal viewpoint, and as mentioned previously when discussing the issue of safety responsibility, the robot is a product and so is not liable for its actions, rather its manufacturer is. This is because, in a failure situation, standard product liability law would apply to the robot, which would cause the person who either set up the device improperly or erroneously or who failed to supervise and stop it before any harm or injury was caused to be held responsible. Frequently, manufacturers claim the error was only human, even when dealing with autonomous systems [31], [33]. Furthermore, in order for a robot to be responsible for the safety issues which can arise as part of a human-robot interaction, it would need to be able to account for its actions and decisions the same way humans do [30]. The alternative to the first law states that it is developers who need to take responsibility for the consequences of errors and failures in human-robot systems. Standards should be created to this end, similar to those in application in the aerospace, medical, and chemical industries [30]. From a technical viewpoint, it is important to add network and physical safety elements, as robots could be hacked into and used for malicious purposes.

Asimov's second law makes the robot a subordinate of the human in almost all situations and so gives way to manipulation of robots for purposes such as hacking or criminal offenses. Instead, the alternative law proposes that communication between a human and the robot be based on the relationship of the roles each has in a given context. The robot can then refuse to execute a request when the human does not have the necessary authority in that situation. This way, the robot will be able to ignore a hacker with malicious intentions, ask the human to confirm his order with a superior or simply issue a warning saying the request is a violation of certain laws or rules [30].

The third alternative law stresses the need for smooth transfer of control from whatever type of relationship between the roles at one point to a new control relationship, no matter what is the nature of an eventual disruption or failure. Asimov's law ignores the dynamics and complexity of relationships between robots and people, how these relationships are expressed and what happens when they need to change rapidly in an unknown environment. Problems with transfer of control have always been present in human interaction with automation and can thus cause various failures [34]. Transfer of control from autopilot to pilot and vice versa is a relevant research area which can inform the design of such systems, as well as human-out-of-the-loop control problems. The concept applies to situations where humans should not have complete control of the robot. For example, when short reaction times are required, the human may not be allowed to override some commands issued by the robot as is the case with autopilot aircraft control. In order to achieve this, designers and developers need to address what is the appropriate situated autonomy and to provide mechanisms permitting smooth transfer of control. This will allow the robot to identify situations when it is better informed than the human, considering latency, sensing, and other variables [30].

The alternative rules put forth above illustrate important open research questions which affect the design of personal robots and highlight the importance of safety, system responsibility and resilience necessary for all human-robot interactions. These alternatives are human-centred and emphasize the fact that responsibility for failures and errors of such systems fall on the side of the robot's developers or stakeholders. This makes the concept of safety and of proving it a major concern for the field of robotics, with a necessity to investigate reliable and appropriate methods of ensuring safety, and creating a well-formed framework on par with existent standards for industries such as the aerospace, automobile, or medical businesses. The alternatives also stress the need for integrating knowledge from social cognition, cognitive engineering, and resilience engineering into the design of personal robots, in order for them to be able to properly and accurately express relationships and obligations through social roles and improve social interaction with humans [30].

2.1.1 Safety

2.1.1.1 Introduction

The personal robots industry is starting to emerge as more and more research is being conducted within the area of developing personal helpers. They can interact with people in various situations, be it as companions, as nursing robots helping people during recovery, as robots helping disabled people or autistic children, or as devices performing dull tasks in order for humans to be able to focus on the more challenging aspects. A pre-requisite for these robots to be developed, manufactured, sold, and be used by humans with confidence, is that they are safe and trustworthy [27].

Robotic assistants are being developed within academia and industry, designed to help humans in many ways. A robot can fetch you a glass of water, or help you stay on course with your diet, learn to interact with you in the long-run by adapting to your personality and needs, or it can perform dull tasks around the house. The multitude of projects studying different aspects and problems of this field is proof that researchers are seeing the huge potential of personal robots and the immense benefits they could have for humans if deployed widely.

An example is the recent project Robot-Era, a European project which aims to develop, implement, and test multiple robotic platforms for interaction and cooperation with elderly people in order to improve their quality of life and assist them [35]. The project is aimed at the ageing population, as recent population projections have shown that the number of elderly people living in Europe will quickly increase in the following years [36]. The project will focus on important challenges faced by robotics, such as cognitive-inspired robot learning architectures, design for acceptability and legal/insurance regulations and standards for real deployment. This, once again, shows the key role played by the necessity of demonstrating robots can pass certification and are safe and trustworthy, especially within an "ageing well" context, when dealing with elderly people.

Another example is the CHRIS (Cooperative Human Robot Interaction Systems) project [37], which brought together more than thirty researchers from four European countries, to address the fundamental issues of safe HRI. The BRL, which was involved in this project until its completion in May 2012, was concerned with human robot co-working, in the context of the human assembling a simple table with a robot. Safety has been studied from the behavioural and physical perspectives within this project, however the issue of verifying the safety of the interactions has not been tackled at the proper level.

How can researchers enhance robots so that they can participate in sophisticated interactions with humans in a safe and trustworthy manner? [27] Close-proximity human-robot interactions require this fundamental research question to be answered and a coherent and credible safety framework to be put in place. Research into the safety aspect concerning high-level behaviours of personal robots during their interaction with humans has not yet been tackled at an appropriate level, although the need for such investigations is openly recognized. High-level behaviour safety analysis implies studying the knowledge of the robot at any moment in time and the mechanisms it uses to makes its decisions at that moment.

Most robotic software architectures use a layered approach. "A robot acting in the real world must use flexible plans because actions will sometimes fail to produce desired effects, and unexpected events will sometimes demand the robot shift its attention. A plan is usually construed as a list of primitive robot actions to be executed one after another but in a complex domain, a plan must be structured to cope effectively with the myriad of unpredictable details it will encounter during execution." [38] "Layers of control system are built to let the robot operate at increasing levels of competence. Layers are made up of asynchronous modules that communicate over low-bandwidth channels. Each module is an instance of a fairly simple computational machine. Higher-level layers can subsume the roles of lower levels by suppressing their outputs. However, lower levels continue to function as higher levels are added. The result is a robust and flexible robot control system. " [39] "The three-layer architecture arises from the empirical observation that effective algorithms for controlling mobile robots tend to fall into three distinct categories: 1) reactive control algorithms which map sensors directly onto actuators with little or no internal state, 2) algorithms for governing routine sequences of activity which rely extensively on internal state but perform no search, and 3) time consuming (relative to the rate of change of the environment) search-based algorithms such as planners. " [40]

The quotes above reinforce the idea that low-level layers generally deal with reactions and control systems, while high-level ones deal with the robot's knowledge, goals and plans. Since the focus of safety analysis so far has been on the lower levels, it is the higher ones that need to be addressed at present, i.e. the levels where the robot takes its decisions. Decision making which takes place in these higher levels is abstracted as an agent. Computer systems capable of independent, autonomous action to meet their objectives are called rational agents, and they can decide what action to take on their own in a given situation [41]. When considering humans, intentions are treated as elements of incomplete, hierarchical partial plans of action. Although the execution of plans involves present-directed intentions, the plans themselves involve future-directed intentions and are reached through deliberation, perhaps in several occasions [42: 2-3]. When considering a rational agent, its decisions are based on changing motivations, which depend on the agent's selected goals and the courses of action it will take in order to achieve them. Thus, the decisions an agent makes is "rational", meaning it should be reasonable, justifiable, and explainable [43]. Developing rational agents which can reason and plan in a continuously changing environment, are capable of goal-oriented reasoning, and react quickly to unforeseen changes in the environment is a serious focus of research in Artificial Intelligence [44: 972].

2.1.1.2 Verification and Validation

As mentioned in section 1.2 Topic Importance, verification and validation can be defined as follows [27]:

Verification is the process of confirming that a system matches its formal requirements.
Validation is the process of confirming that the system results in the intended behaviour once the system is introduced and integrated in its target environment.

Verification can either be performed formally or by simulation. The former performs verification exhaustively for all cases, i.e. one can prove a property of the system. The latter is selective in terms of the tests that can be run within reasonable time. It uses input vectors and it checks the behaviour of the design when the input vectors are simulated against the specification, i.e. each checker can distinguish correct from incorrect behaviour, as defined in the specification [45: 111 - 112].

Formal (or static) verification carries out a comprehensive mathematical analysis of the potential behaviours within a system, which are specified using logical formulae checked against models of the system in various ways [27]. One of the most widely used methods is automatically checking the specification against all possible executions of the systems and is named model checking. It is used for finite-state reactive systems, such as sequential circuit designs and communication protocols. It works by modelling the system as a state-transition graph, expressing the specification in temporal logic, and performing a search procedure in order to find out if the graph satisfies the specification [46].

Since model checking is done automatically, tools exist which perform the task. An example of a powerful tool is SPIN, which detects software defects in concurrent system designs and has been used for control software of interplanetary spacecraft [47]. The method and mathematical background required to understand it are also described by Huth and Ryan in [48]. Further references which show that model checking has been successfully implemented can be found in [49] and [50]. The former is the SLAM toolkit which checks safety properties of software without the need for user-supplied annotations or abstractions, and the latter presents a new verification technique for complex hardware devices that allows generality and a high degree of automation.

A further important research area is agent-oriented formal verification. Bordini et. al. [51] describes a model checking technique implemented in a logic-based, agent-oriented programming language, automatically verifying agent systems and reducing any errors which might be introduced by checking the model of the system (which is usually the method applied by developers) rather than the system implementation. In [52] verification of multi-agent systems is presented, a technique based on model checking through ordered binary decision diagrams. Other work on agent systems involves adapting model checking methods to the formal verification of such systems. A layer of abstraction has been developed in [53], which sits between the verification system and the agent programming language (in this case, the approach is used for programming languages based on the MDI, namely belief - desire - intention, model of agency), through which the agent programming language can be verified. Another example is the design of an intermediate language for BDI style programming languages [54]. Besides allowing verification of what the agent can do, these methods also permit verification of why the agent chooses to do so. The aim is for formal verification to assess all possible motivations and choices, in an exhaustive manner [27].

Because of the exhaustive nature of formal verification, models of the environment or of the situation are used when dealing with complex systems. Even if the system proves to be correct as a result of verification, there exists the issue that the specification can be incomplete and not cover all possible behaviours of the system [45]. This is the reason why, in practice, testing methods are used to assess a system's behaviour as completely as possible. The level of coverage of this behaviour is essential and the aim is to achieve a high level of coverage. Testing methods can use Monte-Carlo techniques and dynamic test refinement to achieve this level. Automating a large part of testing-based verification is also a practice, as formal verification is considered to be complementary to simulation-based techniques [55: 14]. Functional correctness can also be tackled by using assertion-based methodologies and property checking techniques, which provide a higher level of correctness confidence. [56: 1]. The resulting methodology for verification is thus a mix of techniques and is quite robust in practice, being widely used in the design of both microelectronic systems and avionics systems [27].

Simulation-based (or dynamic) verification works by applying stimulus to a software model of the design. The design is correct if its behaviour is as expected for all possible input vectors. The drawback of this method is that it is just a heuristic replacing the infeasible task of checking all possible input vectors for a system. Even if the method aims to be as exhaustive as possible by checking as many input vectors as possible and especially by checking important values (such as for corner cases), it is clear that the level of coverage of these input sequences is extremely important. As such, coverage metrics have been extensively studied within the simulation-based verification community. They essentially measure how much of the design has been tested up to a specific moment in time, what input sequences have been used up to that point, and which other sequences should be used from that moment on in order to explore other important areas of the design [45: 112]. Because coverage within the formal verification context is less clear (even if the design is proven to be correct, the specification could still be wrong), metrics used in dynamic verification can be adapted to suit static verification [45: 113], thus proving that the two methods complement each other and can benefit from combining core ideas.

2.1.1.2 Related Work on Safety in Robotics

Work on safety in Robotics is still considered novel territory. When considering the verification aspect discussed in detail above, it is easy to understand how adaptive systems present a real challenge for proving their safety. Furthermore, the liveness or effectiveness of a system is in competition with the safety aspect, and so an appropriate balance needs to exist between the two.

The initial HRI set-up presented in [2] deals with finding a compromise between safety and verifiability. It identifies safety and liveness properties which are key to the trustworthiness of a robot, and implements a system which respects these properties. Safety means ensuring that no action which is not supposed to be taken by the robot will happen (e.g. the robot will not release the cup when the user is not looking at it), while liveness refers to the robot's usefulness (e.g. the robot will eventually either release the drink, or not release it because it is not safe to do so, and ask the user if he or she wants another drink). A summary of the properties, as identified in [2] are as follows:

(S1) The robot will not release a cup of water unless three conditions are satisfied: the user is looking at the cup, his or her wrist is within close proximity to the cup, and the user is applying enough pressure on the object.

(S2) The robot will not release a cup of coffee (for which the level of associated danger is higher than for water) unless a pre-condition is satisfied: the user is initially looking at the cup, the three core conditions are satisfied as previously, and a post condition is satisfied: verbal confirmation from the user.

(S3) The system does not ask if it should offer another cup unless either all the conditions for releasing the current cup are satisfied and the cup is released or else, if the conditions are not satisfied, the cup is not released and the robot arm returns to the initial position.

(L1) Provided the user complies with the hand-over protocol (this ensures the safety aspect and is hence a safety constraint on the robot's liveness), then the drink will eventually be released. This ensures that the algorithm will, indeed, at some point release the cup.

(L2) When the pre-conditions, conditions, or post-conditions are not satisfied, the robot repeats those steps a maximum number of three times and eventually either releases the drink (more likely if water) or not (less likely if coffee) but, importantly for liveness, then asks the user: "Do you want another cup?"

These properties illustrate the HRI scenario which is similar to that used in the I Robot, I Think project. The former considers two different choices (water and coffee) in order to highlight the possibility of adding or removing conditions depending on the level of danger associated with the object being handed over (for example, the system would be much more restrictive if the robot would be working in a laboratory, handling dangerous chemical substances and safety conditions would have precedence over liveness properties). The latter focuses on the general safety aspect of a handover, building a model which includes taking into account the user's intentions and reactions to the environment. Thus, no matter what the nature of the object being passed is, the robot should be able to take the correct decision of whether to release it or not.

Other work on safety properties uses Model Checking in order to verify a multi-agent robot control system implemented for a robot playing the air hockey game [57: 4809]. This is one of the building blocks in reaching industry-standard safety, as verifiability of a multi-agent control system with learning components has been proved doable [57: 4814].

Work by Bensalem et. al. [58] focused on ensuring safety by construction. It shows that a complex robotic system can be considered as the composition of a small set of atomic components and thus, can be verified [58: 76]. Online safety properties have been also shown enforceable in [59], the authors using a methodology for robotic system modelling and analysis based on the BIP (Behaviours Interactions Priorities) framework, integrated with an existing framework and architecture, the LAAS Architecture for Autonomous System. The aforementioned examples, as well as the ICRA "Formal Methods in Robotics" workshop [60], highlight the fact that the importance of formal methods has been recognized in the Robotics field.

Going back to the agent paradigm, Gordon proposes a method of handling three practical considerations of agents: adaptability to unforeseen conditions, behavioural assurance, and timeliness of agent responses [61: 278]. In order to provide the agents with the ability to adapt and apply formal verification to ensure proper global multi-agent coordination without having the slow nature of verification be a problem, the paper proposes the development of APT agents, agents that are simultaneously adaptive, predictable, and timely. Adaptation is achieved through machine learning/evolutionary algorithms, predictability with formal verification , and timeliness by exploiting the knowledge that learning has occurred to streamline the formal verification. Behavioural assurance following learning is achieved through incremental reverification algorithms, which resemble the idea of local model checking because they localize verification. The method works because the verification methods are tailored specifically to the learning operators used by the machine learning algorithms performed by the agents [61].

Other safety related work includes extending standard failure analysis approaches to address safety analysis for adaptive systems and putting forth a method of increasing the dependability of an adaptive system [62], as well as combining methods from the field of human factors engineering (HFE) and the formal methods community to use model checking with HFE practices in order to formally verify a human-interactive system [63]. Both papers acknowledge the fact that additional modelling tools and technological developments are necessary for proper verification of such systems, and that the research conducted to this end is yet at an incipient stage.

A concept which introduces the HMM theory described in the next subsection is that of "Runtime Verification with State Estimation" [64]. Runtime verification (RV) is defined as deciding whether an execution trace of a program satisfies a temporal logic formula (a way of expressing RV specifications) of the program. In situations in which monitoring overhead is reduced by sampling, there may appear gaps in the observed program executions, making the estimation of the probability that a temporal property is satisfied by a run of the program problematic. The specific problem for which the algorithm was applied was a software model of a planetary rover mission, for which commands are issued from ground to the rover, which then reports the commands to a logger and then to the proper instrument for execution. The status of the execution should be recorded (success or fail), but sometimes the command is simply lost for some reason. The gaps in the sequence of observations (status of a command) are dealt with by using an extension of the Hidden Markov Model of the monitored program to "fill in" these gaps [64]. Thus, the technique allows for estimating the state (determining the probability of a state sequence, given an observation sequence) in situations where the communication channels can be faulty or where the system needs to be able to deal with possible loss of data.

The background concerning safety discussed above clearly highlights the importance and relevance of coming up with dependable verification and validation techniques within the field of Robotics in order to satisfy industry standards for safety. Although formal methods of verification exist and are being successfully used in other well-established industries, extending and adapting them to robots which need to interact with humans is an extremely challenging task. Furthermore, the issue of validation has not been tackled within the context of ensuring that the behaviour of the robot is indeed the intended one once the robot is deployed in its environment. This remains an open question both in academia and industry, and represents an issue which will need to be solved before the widespread use of personal robots can become a reality.

2.1.2 Hidden Markov Models

The theory and concepts behind Hidden Markov Models are explained in detail by Rabiner in [65]. The discussion below follows the explanations, examples, and figures from the paper.

2.1.2.1 Introduction

Real-world process outputs can be characterized as signals, be it discrete or continuous, and having a signal source which can be either stationary (properties are invariable with time), or non-stationary (properties vary over time). Signals can be characterized by constructing signal models, which is an invaluable method for allowing us to learn a great deal about the signal source, without having it available. Signal models have proven to work very well in practice, allowing the realizations of systems such as prediction systems, recognition systems, identification systems, etc., in an efficient manner. [65: 257]

Signal models can be broadly categorized as deterministic (they exploit some known properties of the signal) and statistical (the signal is characterized as a parametric random process, with the parameters of the stochastic process being estimated in a well-defined manner). The Hidden Markov Model is a stochastic signal model and is described below. [65: 257]

2.1.2.2 Markov Chains, Hidden States, and Extension to HMMs

A Markov Chain can be described as a system which can be in one of a set of N distinct states, $S_1, S_2, \ldots, S_N$ at any time and which undergoes a change of state at regularly spaced discrete times according to a set of probabilities associated with each state. An example of such a chain with N = 5 states is illustrated in Figure 1. A state of the system is a Markov state if it only depends on its predecessor state and is conditionally independent on all the other previous states:

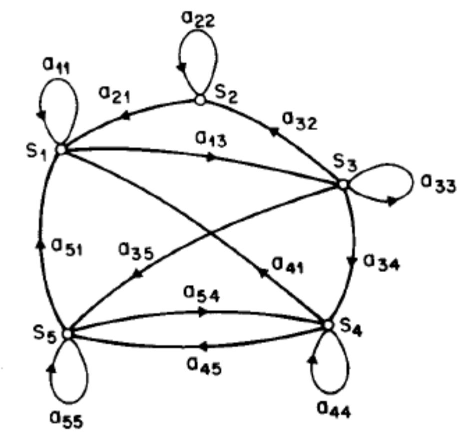


Figure 1. A Markov Chain with 5 states (labeled S1 to S5) with selected state transitions.

$P[q_t = S_j \,|q_{t-1} = S_i, q_{t-2} = S_k, \ldots] = P[q_t = S_j \,|q_{t-1} = S_i],$

where q_t represents the state at time t, and the time instants associated with state changes are denoted as t = 1, 2, ...

This leads to the set of state transition probabilities a_{ij} of the form:

$a_{ij} = P[q_t = S_j \,|q_{t-1} = S_i],\ 1 \leq i, j \leq N,$

with the state transition coefficients having the properties:

$a_{ij} \geq 0 \text{ and } \sum_{j=1}^{N} a_{ij} = 1,$

since they obey standard stochastic constraints.

The above example represents an observable Markov model since the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event, e.g. the weather can be modelled as being rainy, cloudy, or sunny on day t and is thus characterized in one day as being in one of these three states [65: 258].

Modelling a process such that each state corresponds to an observable event is far too restrictive to be applied to real-world scenarios. Extending this model to cases where the observation is actually a probabilistic function of the state engenders the Hidden Markov Model. An HMM is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations [65: 259].

In order to illustrate the concept of an HMM, coin tossing experiments can be modelled. The experiments consist in someone tossing a coin (or several coins) behind a curtain and the only information for the observer is the result of an experiment, i.e. heads or tails. An example of such an observation sequence after a series of **hidden** coin tossing experiments has been performed, is $O = O_1, O_2, O_3, \ldots, O_T = H, H, T, \ldots, T$, where H stands for heads and T stands for tails. An HMM can then be built in order to explain the observed sequence of heads and tails (this involves specifying what the model states represent, as well as the number of states of the model). Three examples of HMMs which could explain a given sequence are given below, as illustrated in Figure 2 [65: 260].

Figure 2 (a) shows a simple 2-state HMM which assumes that one single biased coin is being tossed and which associates each state with either heads or tails (this model would only need a specification for the bias of the coin, i.e. the probability of the coin toss result being heads and the probability of it being tails). This model is equivalent to a 1-state HMM in which the state would represent the coin, and the unknown parameter would represent the bias of the coin.

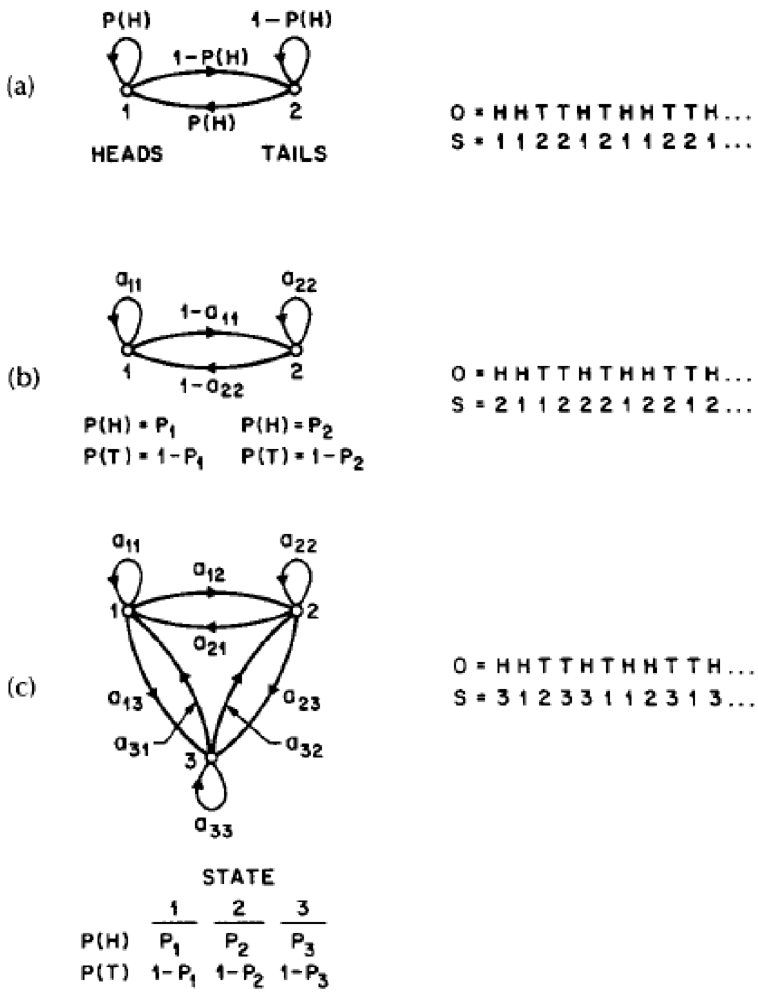


Figure 2. Three possible HMMs which can explain the observed results of hidden coin tossing experiments. (a) 1-coin model. (b) 2-coin model. (c) 3-coin model.

Figure 2 (b) shows a 2-state HMM which associates each state with a different, biased coin, characterizes each state by a probability distribution of heads and tails, and characterizes the transitions between states by a state transition matrix (the mechanism that would explain the state transitions can be itself based on a probabilistic event).

Figure 2 (c) shows a 3-state HMM which is similar to the 2-state model, but corresponds to using three biased coins and choosing one of these, based on some probabilistic event.

Choosing a simpler model has the advantage of necessitating fewer unknown parameters, but a more complex one has a greater degree of freedom and can be more capable of representing complex systems, although limitations on the size of a model need to exist for practical considerations. The choice of an appropriate HMM to model a system is not an easy task, considering that the best one is that which is closest to the real mechanism creating the observations, and that over-specification or under-specification would not produce good results.

The formal definitions of the elements of an HMM are as follows:

- N = The number of states in the model. Individual states are denoted as $S = \{S_1, S_2, ..., S_N\}$, and the state at time t is denoted as q_t.
- M = The number of distinct observation symbols per state (the discrete alphabet size). The observation symbols correspond to the output of the system (heads or tails in the above example). Individual symbols are denoted as $V = \{V_1, V_2, ..., V_M\}$.
- The state transition probability distribution: $A = \{a_{ij}\}$, $a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i],\ 1 \leq i, j \leq N$.
- The observation symbol probability distribution in state j: $B = \{b_j(k)\}$, $b_j(k) = P[V_k\ at\ t \mid q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M$.
- The initial state distribution: $\pi = \{\pi_i\},\ \pi_i = P[q_1 = S_i],\ 1 \leq i \leq N$.

2.1.2.3 Problem 1 (The Evaluation Problem)

The first problem of interest in real-world applications concerning HMMs is the evaluation problem: how to compute the probability that a sequence of observations was produced by a certain model, or put in different words, how to score how well a certain model matches a certain sequence of observations. The latter point of view helps in situations with contending models to choose the one which best fits the observations. Formally, this is the problem of how to efficiently compute $P(O \mid \lambda)$, given an observation sequence $O = O_1 O_2 \ldots O_T$, and a model $\lambda = (A, B, \pi)$.

The solution for this problem is the first part of an algorithm called the Forward-Backward Procedure. The algorithm defines a forward variable $\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i \mid \lambda)$. This represents the probability of the partial observation sequence $O_1O_2 \ldots O_t$ (until time t) and state S_i at time t, considering the model λ. $\alpha_t(i)$ can be solved for inductively, following the next steps:

- Initialisation: $\alpha_t(i) = \pi_i b_i(O_1),\ 1 \leq i \leq N$.
  This step initialises the forward probabilities as the joint probability of state S_i and the initial observation O_1.
- Induction: $\alpha_{t+1}(j) = \left[\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1}),\ 1 \leq t \leq T-1, 1 \leq j \leq N$.
  $\alpha_t(i)$ is the probability of the joint event that $O_1 O_2 \ldots O_t$ are observed, and that the state at time t is S_i, so the $\alpha_t(i)\, a_{ij}$ product represents the probability of the joint event that $O_1 O_2 \ldots O_t$ are observed, and that state S_j is reached at time t+1 through state S_i at time t. Thus, summing this product over all the possible N states S_i at time t, gives us the probability of S_j at time t+1with all the previous partial observations. In order to then find $\alpha_{t+1}(j)$, the sum is multiplied by the probability which accounts for observation O_{t+1} in state j, namely $b_j(O_{t+1})$. The computation is performed for all states j, $1 \leq j \leq N$, for a given t, and is then iterated for $t = 1, 2, \ldots, T-1$.

- Termination: $P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$.
  This step gives the desired calculation, since $\alpha_T\ (i) = P(O_1 O_2 \dots O_T, q_T = S_i \mid \lambda)$.

$\alpha_t\ (j), 1 \leq t \leq T, 1 \leq j \leq N$ requires on the order of N^2T calculations, rather than $2TN^T$ required by the direct calculation method of enumerating every possible state sequence of length T, which makes the Forward-Backward procedure quite a bit more efficient.

Figure 3 shows how state S_j can be reached at time t+1 from the N possible states S_i at time t, and figure 4 shows the lattice upon which the forward probability calculation is based (since there are only N states, all possible state sequences will remerge into these N nodes, no matter how long the observation sequence is).

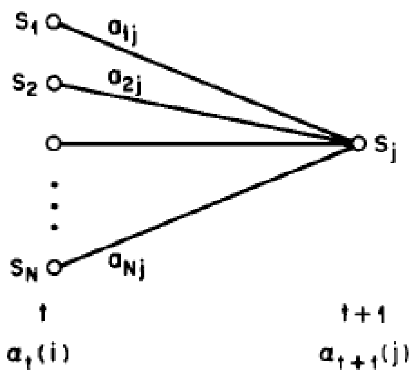


Figure 3. Illustration of how state S_j can be reached.

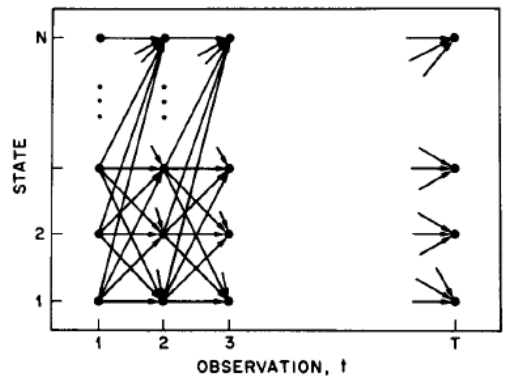


Figure 4. Implementation of the computation of $\alpha_t(i)$, based on a lattice of observations t, and states i.

Only the Forward procedure is needed in order to solve this problem of interest, but the Backward procedure is introduced here due to its similarity, and will be used to solve Problem 3.

The Backward Procedure defines a backward variable $\beta_t\ (i) = P(O_{t+1}\, O_{t+2} \dots O_T \mid q_t = S_i, \lambda)$, which represents the probability of the partial observation sequence from t+1 to the end, given state S_i at time t, and the model λ. $\beta_t\ (i)$ can be solved for inductively as follows:

- Initialisation: $\beta_T\ (i) = 1, 1 \leq i \leq N$.
  This step initialises $\beta_T\ (i)$ to 1 arbitrarily, for all i.
- Induction: $\beta_t\ (i) = \sum_{j=1}^{N} a_{ij}\, b_j\ (O_{t+1})\, \beta_{t+1}(j),\ t = T-1, T-2, \dots, 1; 1 \leq i \leq$ N.
  All possible states S_j at time t+1, the O_{t+1} observation in state j, and the remaining partial observation sequence from state j, all have to be taken into account in order to have been in state S_i at time t (Figure 5).

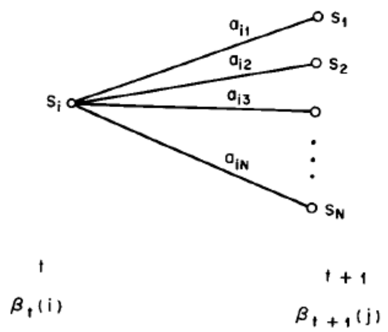


Figure 5. Illustration of the sequence of operations required for the computation of the backward variable $\boldsymbol{\beta_t}$ $(\boldsymbol{i})$.

Like the Forward procedure, the Backward procedure requires N^2T calculations, and can be computed in a similar lattice structure. [65: 262-263]

2.1.2.4 Problem 2 (Hidden State Recovery)

The second problem of interest for HMMs is finding the "correct" hidden state sequence. Since HMMs only assume what the hidden states of a system are, there is actually no absolute, correct hidden state sequence, instead this problem is solved by finding a sequence which is optimal in some meaningful sense (the one that best explains the observations). Formally defined, the $Q = q_1 q_2 \ldots q_T$ sequence needs to be chosen, so that given the observation sequence $O = O_1 O_2 \ldots O_T$ and the model λ, Q would represent the "correct" sequence.

The solution for this problem implies defining what an optimal state sequence is, i.e. choose an optimality criteria. One possibility is to choose to maximise the expected number of correct individual states. However, this has the issue of ending up with unfeasible sequences of states when the HMM has state transition probabilities equal to zero, since the method only computes the most likely state at one point in time, it does not take into account the probability of occurrence of sequences of states. A widely used method is the Viterbi algorithm, based on dynamic programming, which maximises $P(Q, O|\,\lambda)$ and thus finds the single best state sequence. The quantity $\delta_{t+1}\,(i) = \max\limits_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_t = i, O_1 O_2 \ldots O_t\,|\lambda$ is defined, representing the highest probability along a single path, at time t, which accounts for the first t observations and ends in state S_j. This quantity is computed by induction: $\delta_{t+1}\,(j) = \left[\max\limits_{i}\ \delta_t\,(i)\,a_{ij}\right] \cdot\ b_j\,(O_{t+1})$. The algorithm is described in the following steps, with ψ representing the array which contains the argument which maximises δ:

- Initialisation: $\delta_1\,(i) = \ \pi_i\,b_i(O_1),\ 1 \le i \le N;\ \ \psi_1\,(i) = 0.$
- Recursion: $\delta_t\,(j) = \max\limits_{1 \le i \le N} \left[\delta_{t-1}\,(i)\,a_{ij}\right] b_j\,(O_t),\ \ 2 \le t \le T, 1 \le j \le N,$

  $$\psi_t\,(j) = \operatorname*{argmax}_{1 \le i \le N} \left[\delta_{t-1}\,(i)\,a_{ij}\right],\ \ 2 \le t \le T, 1 \le j \le N.$$
- Termination: $P^* = \max\limits_{1 \le i \le N} [\delta_T\,(i)\,]$, (the star notation refers to optimal)

  $$Q_T^* = \operatorname*{argmax}_{1 \le i \le N} [\delta_T\,(i)\,].$$
- Path (state sequence) backtracking: $q_t^* \ = \ \psi_{t+1}(q_{t+1}^*),\ \ t \ = \ T-1, T-2, \ldots, 1.$

The above steps show the similarity of the Viterbi algorithm to the forward calculation, except that it uses maximisation over previous states instead of summation. [65: 264]

2.1.2.5 Problem 3 (Model Parameters Optimisation)

The most difficult problem concerning HMMs is that of optimising the model parameters $\lambda = (A, B, \pi)$ so as to best describe how an observation sequence is generated (maximise $P(O \mid \lambda)$). The observation sequence or sequences used to this end is named a training sequence, because it "trains" the HMM. This problem is essential for real-world applications, since it allows optimal adaptation of the parameters to observed training data.

Since there is no known way to analytically solve for the model which maximises the probability of the observation sequence, the parameters can be chosen such that $P(O \mid \lambda)$ is locally maximised. Techniques using the EM (Expectation - Maximisation) Algorithm [66], [67], or using gradient techniques [68] have been developed, but a widely used method is presented below, named the Baum-Welch Algorithm.

A new quantity is defined, namely $\xi_t\,(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$, representing the probability of being in state S_i at time t, and state S_j at time t+1, given the model and the observation sequence. The conditions required by this quantity are illustrated in Figure 6.

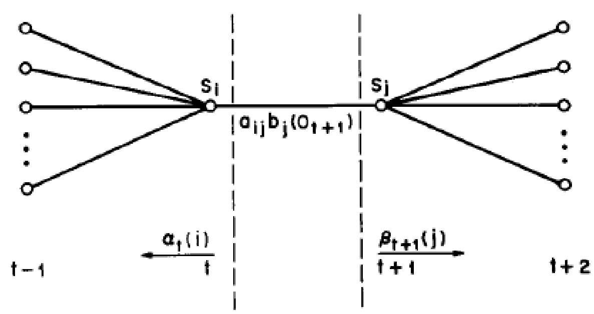


Figure 6. Illustration of the sequence of operations required for the computation of the joint event that the system is in state S_i at time t and state S_j at time t+1.

This newly created quantity can be expressed in terms of the forward and backward probabilities:

$$\xi_t\,(i,j) = \frac{\alpha_t\,(i) a_{ij}\, b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)} =$$

$$= \frac{\alpha_t\,(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t\,(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}$$

where the numerator is the $P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$ term, and the division by $P(O \mid \lambda)$ gives the desired probability measure.

$\gamma_t\,(i) = P(q_t = S_i \mid O, \lambda)$ is also introduced, representing the probability of being in state S_i at time t, given the observation sequence O and the model λ. The quantity can then be expressed as follows:

$$\gamma_t\,(i) = \frac{\alpha_t\,(i)\beta_t\,(i)}{P(O \mid \lambda)} = \frac{\alpha_t\,(i)\beta_t\,(i)}{\sum_{i=1}^{N} \alpha_t\,(i)\beta_t\,(i)} = \sum_{j=1}^{N} \xi_t\,(i,j)$$

We can also make the following observations:

$$\sum_{t=1}^{T-1} \gamma_t\,(i) = \ \ expected\ number\ of\ transitions\ from\ S_i$$

$$\sum_{t=1}^{T-1} \xi_t\,(i,j) = expected\ number\ of\ transitions\ from\ S_i\ to\ S_j$$

The formulas for reestimating the HMM parameters takes into account the above observations and the concept of counting occurrences:

$$\overline{\pi_i} = expectected\ frequency\ (number\ of\ times)\ in\ state\ S_i\ \ at\ time\ (t=1) = \gamma_1\,(i)$$

$$\overline{a_{ij}} = \frac{expected\ number\ of\ transitions\ from\ state\ S_i\ to\ state\ S_j}{expected\ number\ of\ transitions\ from\ state\ S_i} = \frac{\sum_{t=1}^{T-1} \xi_t\,(i,j)}{\sum_{t=1}^{T-1} \gamma_t\,(i)}$$

$$\overline{b_j(k)} = \frac{expected\ number\ of\ times\ in\ state\ j\ and\ observing\ symbol\ V_k}{expected\ number\ of\ times\ in\ state\ j} = \frac{\sum^{T}{}_{\substack{t=1 \\ s.t.\ O_t = V_k}} \gamma_t\,(j)}{\sum_{t=1}^{T} \gamma_t\,(j)}$$

Using the above formulas to reestimate the $\lambda = (A, B, \pi)$ model, we obtain the new model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$, which represents a new model from which the observation sequence is more likely to have been produced. Baker [69] and Baum [70] proved that: either the initial model λ defines a critical point of the likelihood function, in which case $\overline{\lambda} = \lambda$, or $\overline{\lambda}$ is more likely than λ (meaning $(P(O|\overline{\lambda}) > P(O|\lambda))$. Iteratively using $\overline{\lambda}$ instead of λ and repeating the reestimation procedure, the probability of O being observed from the model is improved until some limiting point is reached. [65: 264 - 265]

2.1.2.6 Types of HMMs

A standard type of HMM is named the ergodic model, which has the property of every state being able to be reached from every other state of the HMM and has all a_{ij} coefficients positive. Another useful type of HMM is the left-right model, which is particularly useful for signals which have properties that change over time, such as speech. The model has the property that the state index either stays the same or increases with time, and so the transition coefficients have the property: $a_{ij} = 0,\ j < i$, meaning that the model cannot transition from the current state to state with lower indices. The model also requires that $\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases}$, since the state sequence must begin in state 1 (and end in state N). [65: 266] Other variations and combinations of models are possible, in order to best suit a particular real-world scenario. However, the two mentioned models are the main types of HMMs.

A different type of classifications dichotomizes HMMs into discrete models and continuous models. The former has been presented so far and uses discrete symbols from a finite alphabet, while the latter uses continuous observation densities. The advantage of continuous models is that it can better be applied to scenarios with continuous signals without suffering degradation from having to quantize them via codebooks or other methods. The most general probability density function (pdf) is of the form $b_j\ (O) = \sum_{m=1}^{M} c_{jm}\, \mathfrak{N}\left[O,\ \mu_{jm},\ \ U_{jm}\right],\ \ 1 \leq j \leq$ N, where O is the vector being modelled, c_{jm} is the mixture coefficient for the mth mixture in state j and $\mathfrak{N}$ is any log-concave or elliptically symmetric density [71] (e.g. Gaussian), with mean vector μ_{jm} and covariance matrix U_{jm} for the mth mixture component in state j. [65: 267]

Other interesting types of HMMs include autoregressive models [72], [73], for which the observation vectors are drawn from an autoregression process, models in which the observations are associated with arcs of the model instead of with states [74], and variable duration models [75], [76], which eliminate the conventional HMM weakness of modelling the state duration.

2.1.2.7 HMM Implementation Issues

When implementing an HMM in a real-world scenario, several issues appear, which include scaling, multiple observation sequences, initial parameter estimates, missing data, and the choice of the model size and type.

The issue of **scaling** becomes apparent when considering the definitions of α and β, namely $\alpha_{t+1}(j) = \left[\sum_{j=1}^{N} \alpha_t\ (i)\ a_{ij}\ \right] b_j(O_{t+1}),\ \ 1 \leq t \leq T-1, 1 \leq j \leq N$ and $\beta_t\ (i) = \sum_{j=1}^{N} a_{ij}\ b_j\ (O_{t+1})\ \beta_{t+1}(j),\ \ t = T-1, T-2, \dots, 1; 1 \leq i \leq$ N, respectively. Each term a and b is less than 1, and as t increases, α and β start to head exponentially to 0, which will cause the computation to exceed the precision range of a machine. The scaling procedure depends on the real-world system, but the basic procedure is to multiply $\alpha_t(i)$ and $\beta_t\ (i)$ by scaling coefficients which are independent of i (dependent only on t), such that the end computation will cancel out the coefficients. [65: 272]

Some systems need to use **multiple observation sequences**, either due to the nature of the system, or to the model used. One method of solving this problem is to modify the reestimation formulas by adding together the individual frequencies of occurrence for each sequence. [65: 273]

Because the values obtained from the reestimation formulas presented above should correspond to a local maximum of the likelihood function, one problem is how to choose the **initial parameters** so that this local maximum is the global maximum. A universal method for solving this issue does not exist, but experience has shown that the π and A parameters can be initialised randomly or uniformly, with good results for the reestimates. The B parameters, however, usually require some good initial estimates. There are several methods of obtaining these estimates, which include manual segmentation of the training observation sequences into states with averaging number of observations per state, maximum likelihood segmentation of observations with averaging, k-means segmentations with clustering, etc. [65: 273 - 274]

Insufficient training data in some situations can affect the outcome of the HMM computations. Frequently, observations do not contain sufficient number of occurrences for all symbols within states to give good estimates of the model parameters. Increasing the size of the training set is not necessarily a solution because of practical reasons (the size would have to be increased until there would be a sufficient number of occurrences for all events). Reducing the size of the model is another possibility, but this cannot always be achieved (e.g. when the model is being used for physical reasons). Another solution is to interpolate one set of parameters estimates with another from a model for which more training data exists (a smaller model). [65: 274]

The issue of **choosing a model** when implementing an HMM does not have a straightforward solution. The model type (ergodic, left-right, or some other combination), the model size (number of states), and the observation symbols of the model (discrete or continuous, single or multi-mixture, choice of observation parameters) very much depend on the system which is being modelled and on the type of the signal. [65: 274]

2.1.3 Human-Human and Human-Robot Handover Research

Human-human handovers have not been extensively studied. Single-person tasks of handling objects have been studied by many researchers. However, there are scarce resources when trying to find literature describing the full interaction between two or more people in the context of handing over an object. One of the studies which did investigate human-human handovers is [29], but it only tackles the grip and load forces, and their dynamic. The process of handing over an object from a human to another was also investigated in [77]. The study focuses on the trajectories of the people involved in the process, the point of handing the object from one person to the other, and the velocity pattern of the transfer. It did not tackle any behavioural aspects or visual cues of the people involved in the handover. [78] studies the handover of wooden cubes from a person to another and looks at the mechanisms of coordinating sequences of actions between human subjects, focusing only on the temporal and spatial parameters of the handover.

Human-robot interaction scenarios were investigated, the majority focusing on the kinematics of the motion leading to the handover, with few actually considering the entire interaction of the handover. Shibata et. al. [79], and Huber et. al. [80] worked on implementing human-like trajectories with robots and comparing them with different robot trajectories. Different types of handover modes were investigated in [81] and [82]. Robot posture and gestures during handovers were studied in [83] and it was concluded that some gestures are considered by humans as handover initiation more than others. It was also found that a robot's reaching gesture can be interpreted by humans as a cue to take the object from the robot [84].

There is an even more salient lack of literature concerning visual attention and behavioural elements of a human before and during a handover process. Shic. et. al. [85] argue that the brain allocates movements of the eye to focus on certain locations of a scene which are maximally informative, in order to best use focal visual attention, considered a scarce resource. The core idea suggested is that tapping into the underlying motivation of the human is possible by tracking, recording, and modelling the movements of subjects watching a visual scene [85: 780]. The authors reference Luck et. al. [86]: "The dynamics of foveal fixation represent the allocation of a scarce resource that reflects the changing internal processes, goals, and motivations of the human observer". Even if the precise mechanisms, purpose, and utility of eye movements are not universally agreed on, it is known that internal mental processes affect eye movements [87]. This suggests that the human's visual focus of attention in a handover scenario can indeed be a meaningful representation of the user's intention of engaging in the interaction.

2.1.4 Theory of Mind Concept

"One of the fundamental social skills for humans is the attribution of beliefs, goals, and desires to other people. This set of skills has often been called a <<theory of mind>>." [3: 12] Researchers from many different disciplines have investigated the "theory of mind" concept and have tackled it from various perspectives, including philosophy, ethology, research on the development of social skills in children, research on pervasive developmental disorders (e.g. autism), etc.

Two of the most important models, which take into account elements from multi-disciplinary research, are one from Baron - Cohen [88], and one from Leslie [89]. Investigation into the "theory of mind" concept allows for more complex social skills to be developmentally constructed from simpler sensory-motor skill sets. Building robots which can interact socially with people requires more research into the issues that theory of mind addresses. The advantages of taking this theory into account include having detailed and controlled manipulations of the model while keeping the same environment used for testing human subjects, the ability to vary internal parameters depending on the environmental conditions, and being able to use similar evaluation criteria to those applied to human subjects.

I will next repeat a few ideas mentioned in subection 1.3, as they are part of the background in this area. It is believed that social interaction requires a focus on the issues theory of mind research addresses [3]. The "theory of mind" framework is also used in [28: 311], but from the perspective of the user, researchers applying a "mindreading mechanism" that is indispensible in human-human interactions to a model of human-robot communications. They implemented a robot interface system that applied the proposed model and carried out experiments which upheld the hypothesis that a human can estimate a robot's intention with ease by reading the robot's mind, as well as understanding the robot's unclear utterances made by synthesized speech sounds.

2.2 Current Project

The I Robot, I Think project implements a "personalized" HMM, one which uses multiple observation sequences, uses as input the robot's motor current values, and has as hidden states the main states associated with the human-robot interaction scenario of the robot handing a cup to the user. The implementation uses the core ideas described in subsection 2.1.2, but is adapted so that it fits the system at hand. It uses a version of the Baum-Welch Algorithm in order to implement parameter estimation and the Viterbi Algorithm to find the sequence of hidden states. The in-depth explanation of the implementation is below, in subsection 4.2.1.

The HMM implementation is used with the purpose of the system detecting when the user is grabbing the cup (this represents one of the hidden states of the model). The program can be run at any point during the interaction, and is able to estimate the state of interaction, along with the probability of being in that state. This probability is associated with a degree of confidence of the system that the interaction is in that particular state.

The addition of an extra layer on top of the HMM model takes into account the intentions and reactions of the user. The initial tests performed on the system, backed up by the collaboration with experimental psychologist Dr. Ute Leonards, lead to the implementation of a layer which can be thought of as a subset of the "theory of mind" concept. The robot will always check if the user follows a specific sequence of actions before considering it is safe to release the cup and hand it over. The successful completion of this sequence is interpreted as the intention of the human to grab the cup and the reason behind this is that, when this sequence is indeed followed, the user is engaged in the interaction.

This concept can be used as an initial insight into a method of validation. To remind the reader, validation is the process of confirming that the system results in the intended behaviour once the system is introduced and integrated in its target environment [27]. Taking into account the user's behaviour suggesting he or she is or is not ready for the handover makes the user part of the system and gives the robot essential information about the user's intentions. This represents an important step toward integrating knowledge into the system about the human's next move, and using this to demonstrate the safety of the handover process.

3 Third-Party Hardware and Software

The third-party hardware and software components I used for the execution of this project are mainly based on the hardware and software modules already in place at the Bristol Robotics Laboratory to control BERT2. The software platform used for this purpose was developed by Dr. Alexander Lenz and Dr. Sergey Skachek during the CHRIS project [37].

The third-party hardware components used are as follows:

- BERT2 (Bristol Elumotion Robotic Torso 2): a humanoid upper-body robot from the BRL capable of complex motions, resembling a human's.
- VICON: a motion capture (MoCap) system which detects and localises interaction objects and the human's body parts in 3D space, by the use of retro-reflective markers.
- A hat fitted with retro-reflective markers which is used in order to estimate the human's focus of attention.

Hardware components which have been bought and/or designed specifically for this project include:

- A long-shaped object used as the "cup" in the interaction, which is fitted with retro-reflective markers and copper coating in order for the system to detect when it is being touched by the user wearing a special glove.
- A glove fitted with adhesive copper contacts at the tip of the fingers which helps the system detect exactly when the user wearing it touches the cup.
- An interface kit obtained from the Phidgets [90] web-site, which is programmed to pick up the signal created when the glove makes contact with the cup.

The third-party software components used are as follows:

- Implementation of two databases by Dr. Alex Lenz and Dr. Sergey Skachek which store the objects from the environment. Information about the objects is available via streams to all modules.
- Implementation of a module by Dr. Alex Lenz which allows for bi-directional communication between a module and the voice system.
- Implementation of software by Dr. Alex Lenz which controls the motions of BERT2.
- I used the basic implementation of estimating the human's focus of attention written for a different task by Dr. Alex Lenz as a basis for implementing a solution dealing with estimating the human's focus of attention with respect to the cup being handed over.

4 Project Execution

4.1 The Platform Infrastructure

I developed the project on the Bristol Elumotion Robotic Torso 2 (BERT2) from the Bristol Robotics Laboratory (BRL). BERT2 is an upper-body humanoid robot designed to investigate complex human-robot interaction (HRI) [1]. It is built with the aim of allowing researchers to investigate different aspects of HRI, including verbal and non-verbal communication, gaze and pointing gestures in a real word 3D setting [1]. BERT2 is shown below, in Figure 7.

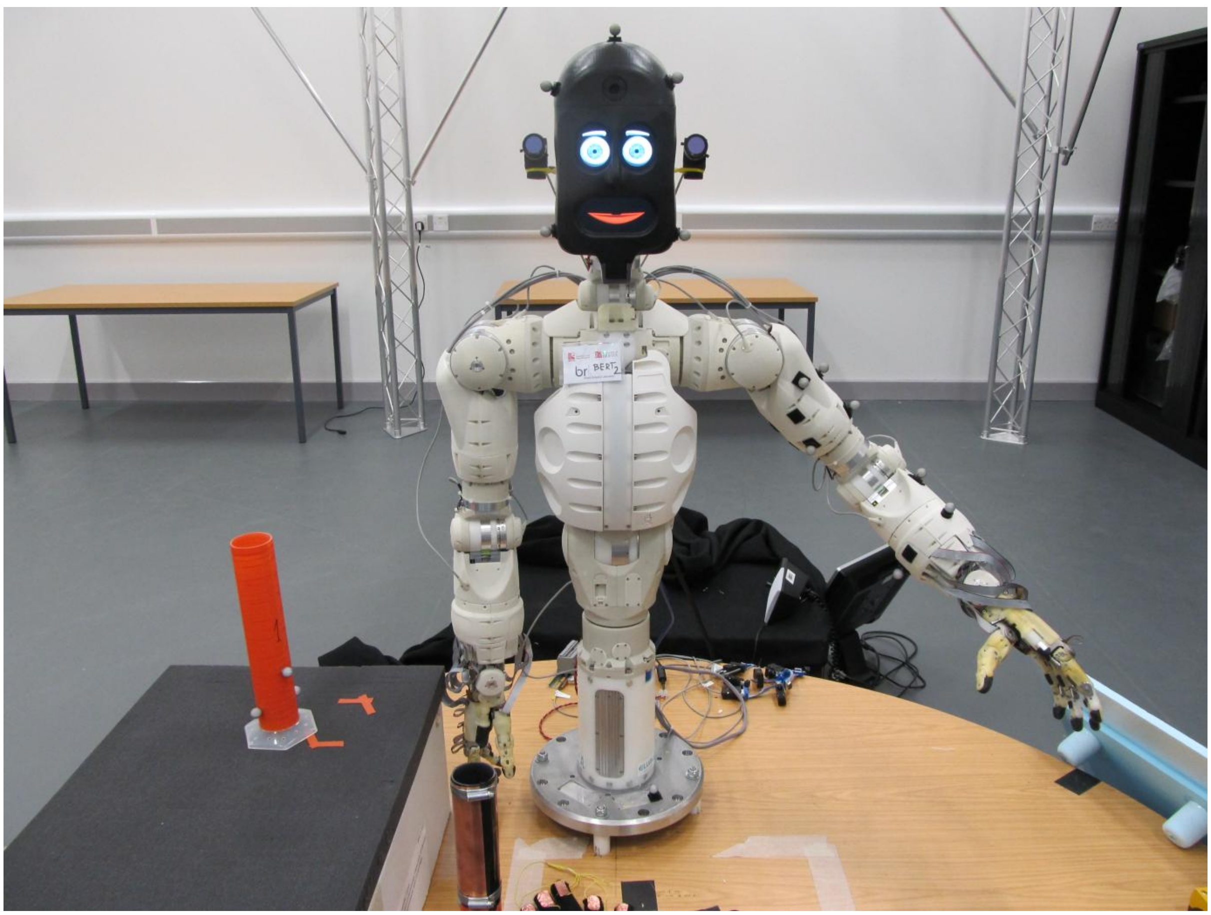


Figure 7. BERT2 in "zero" position.

4.1.1 Hardware

BERT2's torso comprises four joints, namely hip rotation, hip flexion, neck rotation, and neck flexion and its arms are each equipped with seven degrees-of-freedom. This allows for the torso to move backward, forward, twist and for the arms to have movements which closely resemble those of a human, which is important in the HRI context. The joints are controlled through EPOS motor controllers and are also equipped with an absolute position sensor and torque sensor each. The torque value is one of the inputs I used for the HMM implementation which allows for the system to estimate the state of interaction at any moment in time. The joint-level control infrastructure is comprised of the controller area network (CAN) to which all the controllers are connected to, and a central-controller, a Linux PC with a PCI to CAN interface. This central-control provides joint-level error detection and communication monitoring service [1]. BERT2's hands are anthropomorphic and have nine actuators each, controlled in a similar fashion as the arms. The hands can open, close, or grasp very much the same way a human does, which allows for a very realistic study of the object handover scenario.

BERT2's head consists of a plastic moulding combined with embedded colour LCD screens for allowing dynamic expressions of the eyes, eyebrows and lips, as can be observed in Figure 8. This equips the robot's head with the ability of having a wide range of fast changing facial expressions, including eye blinks and saccades [1]. Although this design was thought of with the intent of the robot being able to interact with a human as close to human-human interactions (HHI) as possible, the eyes are not a bi-directional communication channel as is the case in the latter. Thus, the eyes are capable of communicating the direction of attention to the human, but the human's focus of attention is estimated by using the faceLAB head and gaze tracking system from Seeing Machines. It uses an infra-red stereo vision system to detect head pose and position and eye performs tracking by using cornea reflection techniques [1]. The narrow field of view of the tracking technique, however, created the need for me to implement an alternative method of estimating the human's attention, as is described in the Project Execution section. The entire look of BERT2's head was intentionally kept as an artificial, "robotic" one [1], in order to avoid the Uncanny Valley effect, which is a sense of strangeness felt by humans when realizing that a robot which is too similar to a human is indeed just a robot [91], [92].

Figure 8. BERT2's head.

To focus on the subject of HRI without dealing with challenges typically posed by using vision systems mounted on the robot, BERT2 uses the VICON motion capture (MoCap) system to detect and localise interaction objects and the human's body parts in 3D space. The system has sufficient accuracy to follow the motion of human body parts and environmental features and objects using retro-reflective markers [1]. This is the system I used to implement the alternative way of estimating a human's gaze, and thus, focus of attention, as mentioned in the above paragraph.

4.1.2 Software

The computing infrastructure is upheld by YARP (Yet Another Robotic Platform), an open-source project which minimizes the effort devoted to infrastructure-level software development by facilitating code-reuse and modularity [93]. It is an open source C++ library which allows developing independent software modules that communicate with each other using named TCP/IP or UDP channels, named ports (e.g. /humanGazeEstimation/output). These ports can be used as data streaming interface or as remote-procedure-call (RPC) facilities [1]. When integrating a module which performs a specific task with the rest of the system, the development of a YARP wrapper allows all data of interest to be made available to the rest of the system via clearly defined YARP ports.

The robotic system uses two databases to represent the state of the world. The Object Property Database (OPDB) stores static components of all objects present in the world of the interaction scenario. The OPDB can store information (size, color, object's name for spoken language interaction, etc.) about any object which might be present in a scenario (furniture, tools, the robot's body parts, as well as the human's limbs, head, and torso) and it assigns each object a unique identification number which serves as a key across the two databases. Any module can access the information stored in the OPDB as it is implemented as a YARP wrapper around a relational database [1]. The EgoSphere is a fast, dynamic, asynchronous store of object positions (stored in spherical coordinates: radius, azimuth and elevation) and orientations (stored as rotations of the object reference frame about the x, y, z three axes of a right-handed Cartesian world-frame system) and it automatically assigns an EgoSphere ID to an added object. The EgoSphere can be queried by any other module requiring object dynamic information and any further information will need to be gathered from possible additional databases using the object ID assigned by the OPDB. This architecture makes the EgoSphere particularly useful for storing multi-modal information [1].

Besides non-verbal communication, one of the essential components of HRI is enabling spoken language interaction between the robot and the human. BERT2 uses the CSLU Toolkit [93] Rapid Application Development (RAD) which uses the TCL scripting language which allows creating a connection between the actions the robot takes and the spoken dialogue. RAD uses the Festival speech synthesis system and recognition is based on Sphinx-II [1]. The dialogues are constructed via a state-based graphical programming environment. For the cup handover scenario, I created a dialogue in RAD which I then synchronized with the main module (the framework sending information to and from all the other necessary modules). More details about the dialogue can be found in subsection 4.2.

4.2 Implementation

4.2.1 HMM Implementation

I created a model of the system by using a Hidden Markov Model. By breaking down the states in which the system can be in the handover process, I used the following four basic hidden states (N = 4): the robot is grabbing the cup, the robot is holding the cup (without the user touching it), the user is grabbing the cup, and the robot is not holding the cup. Choosing a small number of states allows good differentiation between them and provides the robot with a clear state on which to base the release of the cup, i.e. the "user is grabbing the cup" state.

The chosen model is an ergodic one, meaning that each state from the system can be reached from every other state. I do, however, initialise the transition probability distribution matrix A with some zero values for transition between states in order to encourage the system to reestimate those particular values as close to zero, i.e. a transition from the robot not holding the cup state to the robot holding the cup (without going via the robot is grabbing the cup state) is unlikely, but can happen if the motor current values do not pick up, for various reasons, on the action of the robot grabbing the cup.

The initialisation of the observation symbol probability distribution matrix B is even more important for a correct reestimation of the model parameters. The method I use to solve this problem is to take a representative training sample of observations and segment it into parts corresponding to the four states of the HMM. For each state I then count the number of occurrences of each symbol and this is the value with which that symbol gets initialised for that particular state. This proved to be a sound technique, which provided good results. However, the training sequence used needed to be representative for the handover scenario and incorporate all the usual transitions between states.

In order to choose which input signal to use for the HMM, I recorded several typical handover interactions (5 sets of 20) and plotted both the motor current values from the robot's fingers and the torque values from the robot's arm in MATLAB. The results of a typical handover can be seen in Figure 9. The left subplot shows the motor current values, and the right one shows the torque values. The first vertical blue line represents the moment when the command of grabbing the object was sent to the robot, and the second one represents the moment when the human is grabbing the cup. The change in the values can be clearly observed, especially at these two moments in time. This is how the detection of the robot grabbing the cup and the human grabbing the cup states is achieved. The other two states, the robot holding the object and the robot not holding the object, have relatively constant values. The difference between these values changes for each of the motor current and torque values, and is thus a criteria of selection of which values to use. Based on the plotted graphs of the training data and on experimentation with different signals, including combining some signals (summing the values), the best choice resulted in using the motor current values from the robot's middle finger. This is also intuitive, if we consider the fact that the middle finger is used for every step of the motion of grabbing and releasing the object. The signal can be better observed in Figure 10.

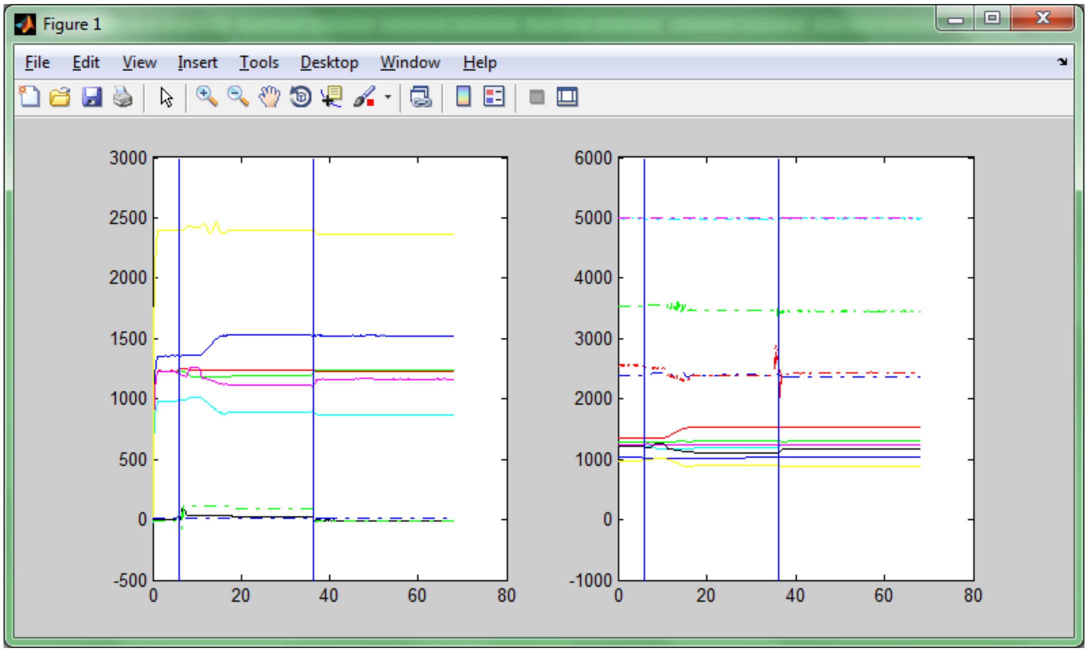


Figure 9. Motor current values (left) and torque values (right) from BERT2 during a typical handover scenario.

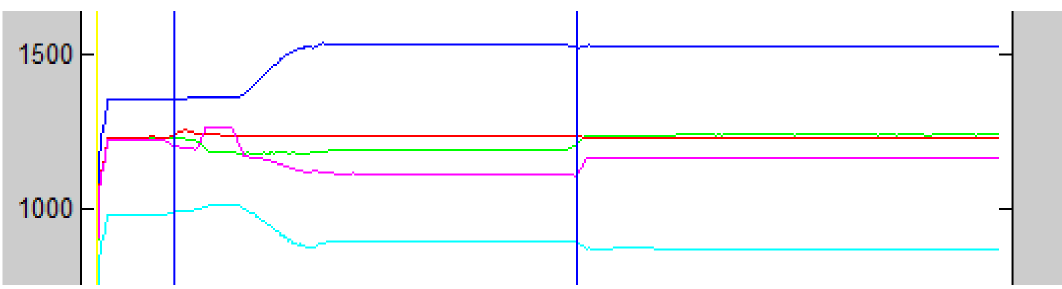


Figure 10. Purple signal represents BERT2's middle finger motor current value.

Based on choosing the stream of values used as input for the HMM, the number of distinct observation symbols per state corresponds to the number of values the stream takes. Throughout the entire experiments I have observed the maximum value as 120 (after considering modifications to transform all the values into positive values) and so M equals 120 in this situation.

The initial state distribution favours the first state (the robot grabbing the cup) in order to provide some structure to the start of each handover, but this does not influence the reestimation values (a uniform distribution can be used instead, with the same results).

The reestimation technique I implemented is based on the Baum-Welch Algorithm, as presented in subsection 2.1.2.5, and using the same formulas. Because of good initial estimates for the model parameters, two iterations of the algorithm are sufficient to reach a satisfactory result. However, in order for the algorithm to actually work, a scaling method had to be implemented. The normalisation technique was implemented following the formulas from [94]:

- Forward Algorithm: $\hat{\alpha}_t(i) = \frac{\pi_i \, b_i\,(O_1)}{\sum_{k=1}^{N} \pi_k \, b_i\,(O_1)}$

$$\hat{\alpha}_{t+1}(i) = \frac{b_i\,(O_{t+1}) \sum_{j=1}^{N} \hat{\alpha}_t(j) a_{ji}}{\sum_{k=1}^{N} b_k\,(O_{t+1}) \sum_{j=1}^{N} \hat{\alpha}_t(j)\, a_{jk}},\ \ 1 \le i < T.$$

- Backward Algorithm: $\hat{\beta}_t\,(i) = \beta_t\,(i)\, \prod_{k=t+1}^{T} \eta_k$, where η_k is the normaliser

$$\hat{\beta}_t\,(i) = \beta_t\,(i) = 1$$

$$\hat{\beta}_t\,(i) = \eta_{t+1} \sum_{j=1}^{N} \hat{\beta}_t\,(j)\, a_{ij}\, b_j\,(O_{t+1})\,, 1 \le t < T\ .$$

- Updating formulas: $\gamma_t\,(i)$ is computed the same way as before because the normalisers cancel out

$$\gamma_t\,(i) = \frac{\alpha_t\,(i)\beta_t\,(i)}{\sum_{i=1}^{N} \alpha_t\,(i)\beta_t\,(i)}$$

$\xi_t\,(i,j)$, however, is computed with the formula:

$$\xi_t\,(i,j) = \frac{\hat{\alpha}_t\,(i)a_{ij}\, b_j(O_{t+1})\eta_{t+1}\, \hat{\beta}_{t+1}(j)}{\sum_{j=1}^{N} \hat{\alpha}_t\,(j)\hat{\beta}_t(j)} = \frac{\gamma_t\,(i)\, a_{ij}\, b_j(O_{t+1})\eta_{t+1}\hat{\beta}_{t+1}(j)}{\hat{\beta}_t\,(i)}$$

The updating formulas for π, a, and b remain the same as before.

The discovery of hidden states was implemented using the Viterbi Algorithm, as described in subsection 2.1.2.4. The formulas are the same, but the implementation uses the log of the values, in order to scale the results. I implemented the algorithm both in MATLAB (to better plot the results), and in C++ (to implement the module which controls BERT2). An example of a determined hidden sequence is in Figure 11. State 1 represents "robot is grabbing the cup", state 2 represents "robot is holding the cup", state 3 represents "human is grabbing the cup", and state 4 represents "robot is not holding the cup". As can be observed, state 3 is of importance in order for the robot to take the decision of releasing the cup. With respect to the probability that the interaction is in this state, experiments have shown that a value of approximately $1.0e + 005\ \cdot (-2)$ represents a good degree of confidence in that state.

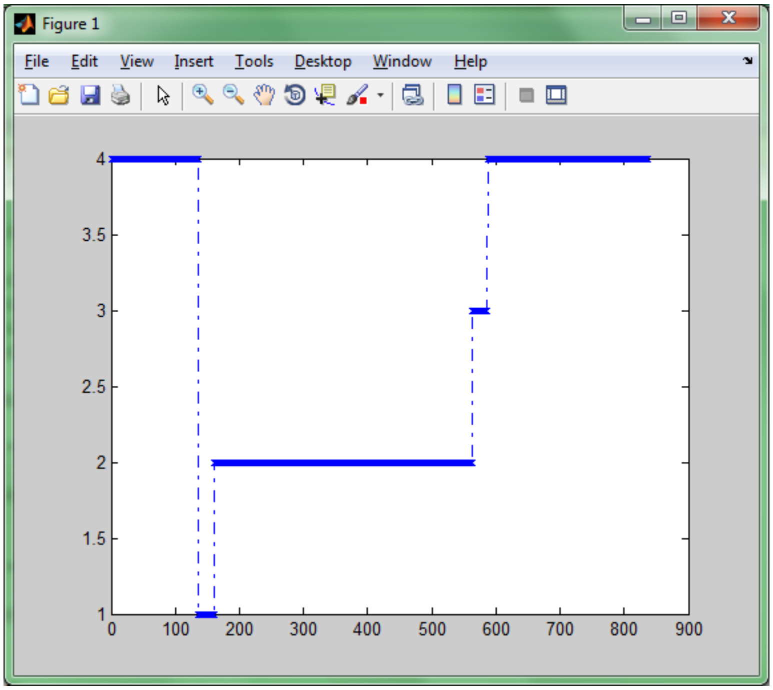


Figure 11. Recovery of hidden sequence. State 1 = "robot is grabbing the cup", state 2 = "robot is holding the cup", state 3 = "human is grabbing the cup", state 4 = "robot is not holding the cup".

Although the implemented algorithm works in most situations, there are some scenarios in which the state is estimated as being "human is grabbing", when it is not the case. This occurs when, for example, the human tries to get the cup, but the robot does not release it for whatever reason. The motor current values after the user releases his or her grip do not go back entirely to their previous values (due to the nature of the robot's motors). In order to prevent this state from being estimated erroneously, I implemented another HMM with two hidden states, one corresponding to the motor current values fluctuating and one corresponding to the values not fluctuating. The idea is that when the human is touching the cup, the values will always fluctuate and so, if the state of the second HMM finds the values constant, it means that the interaction cannot possibly be in the "user is grabbing" state. The implementation of this second HMM is similar to the previous one. The only difference is that the observations it uses are the values of the standard deviations computed for each set of 5 values of the typical observations vector:

$O_1 = \{ O_{11}, O_{21}, O_{31}, O_{41}, O_{51}, O_{61}, O_{71}, O_{81}, ..., O_{N1}\}$ - a typical observation sequence

$O_2 = \{ -, - , -, -, O_{12}, O_{22}, O_{32}, O_{42}, ..., O_{(N-5+1)2} \}$ - a standard deviation observation sequence where O_{12} = standard_deviation ($O_{11}, O_{21}, O_{31}, O_{41}, O_{51}$), O_{13} = standard_deviation ($O_{22}, O_{32}, O_{41}, O_{51}, O_{61}$), etc.

4.2.2 Human Intentions and Reactions Modelling Implementation

In a successful handover scenario, the robot expects the user to perform the following sequence of actions: the user looks away from the object as his or her attention gets caught by different objects or people in the environment, the user looks at the object, the user looks away, the user grabs the object and the handover is completed when the robot lets go of the object. The phase of the human looking away from the cup after looking at it is optional: an extremely engaged user might constantly look at the cup (this is an extreme case). Usually, the next natural action for a human in such a scenario is to look at the robot's head, and then back at the cup. This way, the phase can include a number of alterations of the user's behaviour, as each user's eye movements differs from handover to handover. The system generally detects the user looking to and away from the object several times during it.

The most important element, however, is that the amount of time that passed from the phase when the user first looks at the cup, and the moment he or she actually grabs it does not go above a certain value (this was obtained experimentally). A value which is too low means the user was not paying enough attention to the object, and only looked at the cup briefly, and a value which is too high means that the user lost interest in the object at some point. When the robot detects either one of these cases, it does not consider it safe to handover the cup.

The state machine from the robot's point of view is illustrated below, in Figure 12.

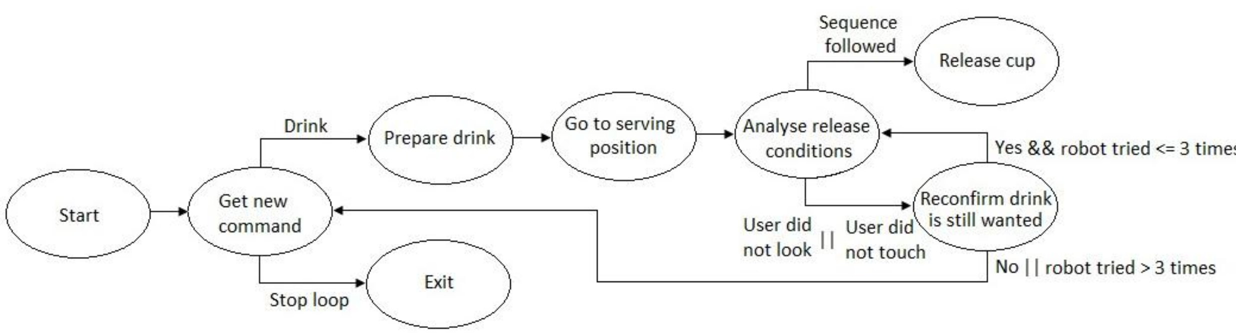


Figure 12. State Machine from the robot's perspective.

As can be seen, the robot asks the user if he or she would like a drink. If the answer is "Yes", it prepares a drink (i.e. it goes to get the cup from a pre-defined pick-up position), goes to a pre-defined serving position, and tells the user to get the cup. At this point, the robot starts checking if the user is following the sequence of actions indicating his engagement in the interaction: [user looks at cup], [user looks away], [user touches cup]. The module which estimates the human's focus of attention works constantly in the background. Therefore, the main module can obtain from it, at any moment in time, the moment when the user started looking at the cup and the moment when he or she stopped looking at it. The only value needed at this point is the difference between the current time and the time at which the user started looking at the object. This basically includes the first two actions in one interval (we are not interested in evaluating the delimitation between these two actions, we are only interested in knowing if too much time has passed since the user first looked at the object). If this value proves to be within the expected range (determined experimentally), the system then checks for the user to be touching the cup via the HMM implementation. I also implemented another alternative for checking when the user is touching the cup. This method is more precise in giving the exact moment when the human touches the object and is used with the purpose of obtaining clearly defined intervals for the above actions. BERT2 considers it safe to handover the object only when the execution of this sequence is complete.

The module which estimates the human's focus of attention is based on an implementation by Dr. Alex Lenz from the BRL for a different task. I adapted the module in order to make it work as described above. It works by streaming information from the VICON motion capture system regarding the position and orientation of the hat fitted with retro-reflective markers, which the user needs to wear during the interaction with BERT2. The module obtains this information from VICON and uses it in order to compute two vectors: one representing the direction of the human's head, and one representing the difference between the human's head and the object of interest. If the difference between these two vectors is small enough, the user's focus of attention is estimated to be on the object. I chose to use this method rather than using the gaze-tracking system and the cameras mounted on the robot's head because of the narrow field of view of the cameras and the limitations of the tracking system (e.g. if the user's eyes are covered by the eyelids due to eye movements, the software fails to properly track the human's gaze). The alternative method proved to be quite robust, its only limitations being that eyes movements which are not accompanied by head movements are not detected this way. In this situation, however, the advantage outweighs the drawback.

The alternative to the HMM implementation for detecting when the user is touching the cup uses a glove fitted with adhesive copper contacts and a long-shaped object acting as the cup in the interaction which is also fitted with copper coating, as well as with retro-reflective markers (for the VICON system to be able to track its position in the environment). I use a Phidget interface kit which I programmed to signal the main module when the user touches the cup. The set-up can be seen in Figure 13.

Figure 13. Set-up for detecting when the user touches the cup.

4.2.3 Verbal Communication

The dialogue between the robot and the user was implemented using the RAD toolkit [93]. I constructed a dialogue via the state-based graphical programming environment which uses the TCL scripting language to programme the actions for each state in the dialogue. The verbal communication between BERT2 and the user is based on the state machine from Figure 12. The dialogue is implemented in such a way that it synchronises with the main module. Each program loops at key points during the interaction, waiting for a trigger from the other (e.g. the main module waits for the voice system to send it the user's answer to the question "Do you want a drink?", the voice system waits for the main module to inform it when the robot has completed the execution of a movement, in order to proceed with the dialogue).

An example of verbal communication for a typical handover scenario is:

BERT2: "Hello. Would you like a drink?"

User: "Yes."

BERT2: "Preparing drink."

[Robot gets drink from the pre-defined pick-up position and takes it to the pre-defined serving position.]

BERT2: "Please get the drink."

[User follows the expected sequence of actions. The robot hands over the cup.]

BERT2: "Enjoy."

In order to illustrate the safety aspect, the following is an example when the user does not follow the expected sequence of actions:

BERT2: "Hello. Would you like a drink?"

User: "Yes."

BERT2: "Preparing drink."

[Robot gets drink from the pre-defined pick-up position and takes it to the pre-defined serving position.]

BERT2: "Please get the drink."

[User does not look at the cup.]

BERT2: "You seem to be distracted. Do you still want the cup?"

User: "Yes."

BERT2: "Please get the drink."

[User follows the expected sequence of actions. The robot hands over the cup.]

BERT2: "Enjoy."

Although non-verbal cues are extremely important in the context of social interaction, when these cues are not enough to ensure the system it is safe to perform a certain action, verbal confirmation is an extra measure which assures the robot tries to steer the interaction toward its intended goal. In this case, the robot is pro-active, and asks the user if he or she is still interested in the drink. BERT2 will repeat this question until it has tried three times and failed, or until the user says "No". If the robot does not consider it is safe to handover the object at this point, it will put the cup back and ask the user if he or she wants a new drink. This ensures the liveness aspect of the interaction, i.e. the robot does not remain stuck in a loop, without doing anything useful. The full dialogue is attached in the Appendix section.

4.2.4 System Overview

All the modules described so far have been implemented in C++. They communicate with each other via YARP, as described in subsection 4.1.2. Figure 14 describes the overview of the software architecture and the dependencies of the implemented modules. Figure 15 is a screen shot of some of the various modules running on the background and providing necessary information about the system.

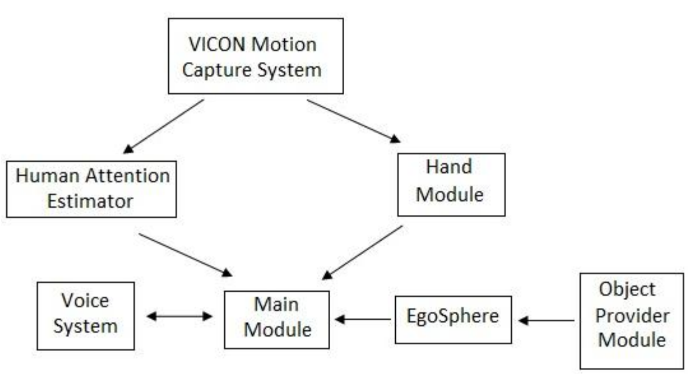


Figure 14. Overview of the software architecture.

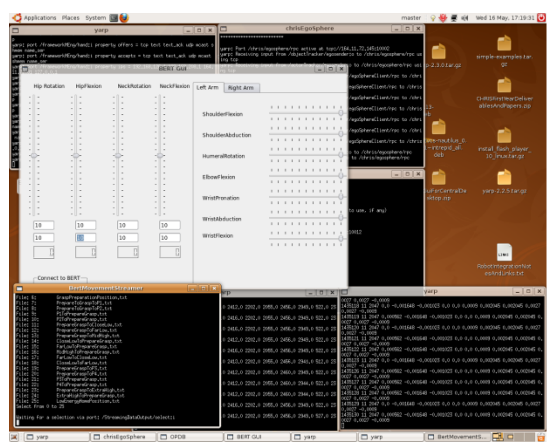


Figure 15. Screen shot with modules running on the background.

4.3 Evaluation

In order to evaluate the system and to compare the basic HMM with the extended system including the model of the user's intentions, I performed a series of experiments at the BRL. The ethical approval necessary for conducting the experiments can be found in the Appendix. The description and results of this evaluation are described below.

4.3.1 Experimental Set-Up

The experiments consisted of users interacting with BERT2, in a drink-serving scenario. Each participant was given an information sheet describing the main purpose of the experiment and giving an overview of how to verbally communicate with the robot, as well as a consent form which needed to be signed. Both these forms are attached in the Appendix.

A user was sat on a chair, in front of the robot, with no restrictions whatsoever on movements. The user was asked to wear the glove described above, as well as the hat with the retro-reflective markers and a microphone. No restrictions were imposed concerning glasses, as the VICON motion capture system was used as an alternative for the gaze-tracking software. The participant was then simply asked to interact with the robot as naturally as possible, knowing that the goal of the interaction was to get a drink from it. The experimental set-up is illustrated in Figure 16.

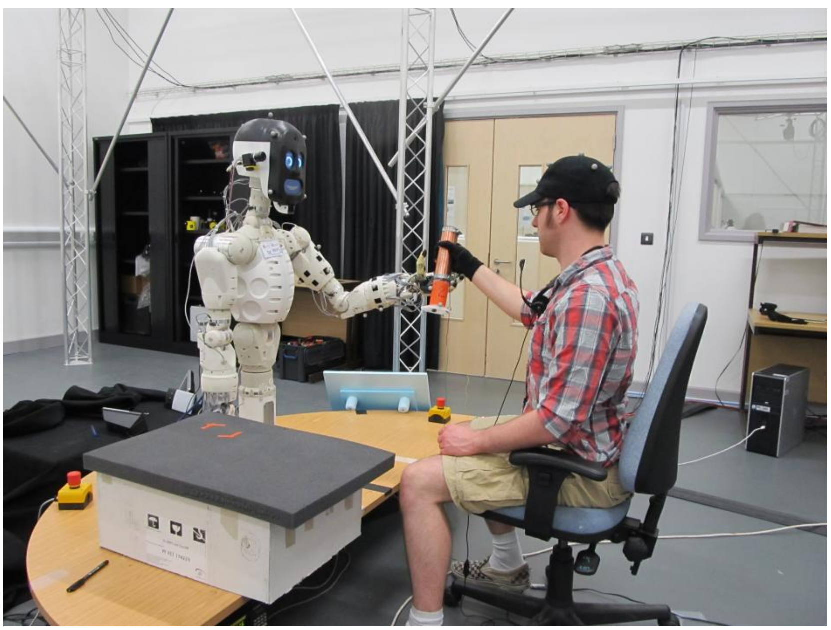


Figure 16. Experimental Set-Up.

The number of participants in the experiments was 17. The subjects included people who were familiar with the BERT2 robotic platform and knew what to expect in terms of the robot's behaviour, as well as people who were naive users and had not interacted with the robot before.

4.3.2 Testing Scenarios

The experiments included three testing scenarios:

- Natural interaction with the robot: the user was simply asked to interact with BERT2 with the goal of getting a drink from it.
- Interaction while engaged in another task: the user was asked to start from 4 and keep adding 7, saying the result out loud, while trying to get the drink from the robot. This scenario simulates a real-world situation in which the user is engaged in a conversation or in other types of tasks, while trying to buy a drink from a robot. It helps with observing how such a task affects the behaviour of the user and how much participants trusted the robot when their attention was divided between it and something else. This is linked to the concept of cognitive load theory [95]. If the user can count successfully, this suggests that the cognitive load of the other task is not too demanding, but if the total cognitive load of both tasks exceeds the capability of the subject, the highest risk task would take priority.
- Surprise distraction: during some of the natural interaction scenarios, a loud noise was played before the moment when the user was supposed to get the cup from the robot. The scenario tested what happens when a user gets distracted suddenly, by an extraneous event. The safety of the handover is particularly important in this case: if the user gets distracted to the point where he or she is not engaged in the handover process any more, the robot should be able to detect it is not safe to release the cup.

The first two types of scenarios contained three runs each, and they were intertwined with a case of surprise distraction (a larger number would not have the intended effect, as the person is already aware a "surprise" might occur and not react to it any more). This constituted a set of experiments. Three sets of experiments were run with each user: one testing the HMM implementation, one testing the HMM implementation extended with the model of the user's intentions, and one testing the model of the user's intentions with the alternative method of checking when the user touched the cup. The users were asked to wear the glove in all cases, to reduce any change in their behaviour between experiment sets. They were only aware of the difference between the natural interaction scenario and the scenario in which they were engaged in the task of counting.

4.3.3 Results

The results did not find a significant difference between the two cases testing the model of the user's intentions, one by using the HMM model and the other by using the glove in order to detect when the participant was touching the cup. The results below will thus refer to the two cases as "the extended model" and will simply add them both in the same category, by considering a set as containing six runs for each of the first two scenarios and two runs for the surprise scenario.

System	Number of successful handovers	Percentage of successful handovers	Number of times cup was not released by the robot	Percentage of times cup was not released by the robot	Number of unsuccessful handovers (cup was dropped)	Percentage of unsuccessful handovers (cup was dropped)	Total number of tries
Basic HMM implementation	74	62.18	34	28.57	11	9.24	119
Extended implementation (user intentions modelling)	180	75.63	55	23.1	3	1.26	238

Figure 17. Comparison between the basic model and the extended model.

The numbers from Figure 17 show a clear difference between the basic and extended models. The percentage of successful handovers is significantly greater in the extended case than in the basic one. The number of drops is reduced for the extended scenario, resulting in only 1.26% of dropped objects, as compared to 9.24% in the basic one. This is because the simple HMM implementation only takes into account values derived internally from the system. It does not consider other elements which might indicate it is not safe to release the cup. Most cases of drops occurred because the motor current values were similar to the values typically encountered in the "user is grabbing the cup" case, but the user was not actually prepared to get the object (he or she was distracted or simply hit the cup without grasping it). From this perspective, the implementation which attributes weight to the user following a specific sequence of actions is safer and reduces false positives.

In the extended case, the percentage of the times the cup was not released by the robot has a higher value than that of the times the cup was dropped. This is to be expected: the robot has a stricter policy to follow in order to release the cup and so emphasizes safety, i.e. it will not release the cup because the user has not looked at it at the right moment. On the other side, when the basic implementation, the robot does not release the cup simply because it does not estimate the state correctly.

In the extended case, the robot only considers it is safe to release the cup when a specific sequence of actions has been followed by the user. The check which corresponds to the participant's focus of attention compares the time passed from the moment when the user looks at the object until he or she touches the cup with a minimum threshold and maximum threshold. If the value is in this range, and the user is still touching the cup, the robot completes the handover. Figure 18 presents the values of the two thresholds obtained from the experiments, as well as the average values for the three types of experiment scenarios, all expressed in seconds.

Scenario	Minimum threshold (s)	Maximum threshold (s)	Average value (s)
Natural interaction	0.3	5	2.3
Interaction while engaged in counting task	0.3	5.8	3.7
Surprise distraction	0.3	6	2.8

Figure 18. Timings for the extended case.

The experiments showed that a user needs to look at the object at least 300 milliseconds before touching the cup in order for the handover to proceed successfully. This minimum threshold is the same throughout scenarios because it represents the minimal amount of time which signifies the participant actually looked at the cup with the intention of being engaged in the handover process.

The maximum threshold in the natural interaction case is 5 seconds, with an average value of 2.3 seconds. The difference in values occurs because users take more or less time to look at the cup and then at the robot, before touching the object. The scenario in which the test subjects are engaged in the counting task results in an increased maximum threshold, as well as a larger average value. This was the anticipated result, as participants have to switch between tasks in order to complete both of them.

The surprise distraction scenario affected participants differently. The average value is lower than that corresponding to the counting task, but higher than that corresponding to the natural interaction. This is because some users reacted stronger than others. When the loud sound was played, some participants would turn their heads and would shift their focus of attention toward the noise, while others would simply go on with the interaction and turn around after the handover was completed. The maximum threshold, however, has a rather high value, which was affected by the cases in which users became disengaged with the handover process.

The results presented above show that the extended system, the one which takes into account the user's intentions behaves safer and more predictable than the basic system, which has no representation of the participant's actions prior to touching the cup. They also give the average, minimum, and maximum values of the timings necessary for incorporating the extended system in a reliable manner.

5 Conclusions and Future Work

5.1 Conclusions

Personal robots can become an important part of people's lives, helping them cope with various situations and performing a wide range of tasks for them. Technology has reached a point where the capabilities of such robots can make the widespread use of robotic assistants a reality. However, a pre-requisite for deployment in the real world is demonstrating that a robot is capable of safe, trustworthy social interaction with humans. From this point of view, robots need to pass certification and be proven safe, and reliable. This is not an easy task for dynamic devices, which adapt their behaviour to fit the task at hand and to cope with changes in their environment. Well-defined methods and standards for demonstrating safety, and reliability need to be put in place in order for personal robots to enter industry.

The I Robot, I Think project investigated the core elements which are needed in order to create a safe interaction between a robot and a human within the context of handing over a drink. The project accomplished its initial objectives:

- It created a model of the human-robot interaction scenario and implemented a method of estimating the state the interaction is in at any moment in time, with an associated confidence level. It did this by means of a Hidden Markov Model and the algorithms associated with this concept.
- It investigated how the user's intentions and reactions to changes in environment affect the robot's decision of releasing the cup by looking into the field of experimental psychology.
- It extended the HMM by adding a layer which models the human's intentions by tapping into the "theory of mind" concept. This means that the robot takes into account its expectations of what the user will do in order to reach the goal of getting the cup. The implementation takes into account the expected sequence of actions a user will make when interested in the handover process. The robot takes the decision of releasing the cup only if the user performs the following sequence of actions: [looks at the cup], [looks away], [touches the cup].
- It implemented an alternative way of detecting the user touched the cup in order to ensure the results of the action timings are consistent.
- It conducted experiments comparing three systems: the basic HMM implementation, the extension of the HMM by adding the "theory of mind" layer, and the implementation of the "theory of mind" layer on top of an alternative way of detecting when the user is touching the cup.
- It showed that the extended system is safer and more reliable than the basic one, and that no matter what the underlying method is of detecting the core information needed by the robot to make its decision (in this

case the moment when the user touches the cup), knowledge about the behaviour of the human and representation of the user's next expected move enhance the system, making it safer and more reliable.

The project is a building block toward demonstrating safety within a human-robot interaction scenario, and can be used for the purpose of validation. This is because the created system takes into account information about the user, who is in fact part of the validation environment.

5.2 Future Work

There exist several ways of enhancing the research performed by the I Robot, I Think project. Working toward creating formal methods of verification and validation is a clear future direction. The project shows how useful statistical models are within the context of human-robot interaction and, so, this is a building block into developing a well-defined proof of safety.

A promising research direction of future work starting from the current project is runtime verification. Moving verification towards verification at runtime, based on extracting information from a running system, can be fruitful effort.

Another extremely interesting line of future work is delving into the "theory of mind" concept and applying an extensive framework to represent the user's beliefs, goals, and desires within a human-robot interaction scenario such as the one studied in this project. Attempting nested levels of representing the "theory of mind" set (i.e. the robot has not only a representation of the user's beliefs, goals, and desires, but it also has a representation of the human's "theory of mind" model with respect to the robot) can prove to be a valuable research direction.

No matter what future directions will be taken from this point onward, the current project constitutes an important insight into essential open research questions which are just starting to be addressed by the robotics community.

6 References

[1] A. Lenz, S. Skachek, K. Hamann, J. Steinwender, A. G. Pipe, and C. Melhuish. "The BERT2 infrastructure: An integrated system for the study of human-robot interaction" in *Humanoid Robots (Humanoids) 10th IEEE-RAS International*, 2010, pp. 346-351.

[2] E.C. Grigore, K. Eder, A. Lenz, S. Skacheck, A. G. Pipe, C. Melhuish. "Towards Safe Human-Robot Interaction" in *12th Conference Towards Autonomous Robotic Systems (TAROS)*, 2011, pp. 323-335.

[3] B. Scassellati. "Theory of Mind for a Humanoid Robot" in *Autonomous Robots*, 2002, vol. 12, pp. 12-24.

[4] School of Experimental Psychology. "Dr. Ute Leonards". Internet: http://www.bris.ac.uk/expsych/people/academic/uteleonards.html, Feb 7, 2012 [April 25, 2012].

[5] P. Dario, E. Guglielmelli, and C. Laschi. "Humanoids and personal robots: Design and experiments", in *Journal of Robotic Systems*, vol. 18, issue 12, 2001, pp. 673-690.

[6] Engineering and Physical Sciences Research Council. "Principles of Robotics". Internet: http://www.epsrc.ac.uk/ourportfolio/themes/engineering/activities/Pages/principlesofrobotics.aspx, Sep, 2010 [April 26, 2012].

[7] H. J. Kelly, V. S. Dan, C. J. John, and R. K. Leanna. "A Practical Tutorial on Modified Condition/Decision Coverage", NASA Langley Technical Report, 2001.

[8] D. Feil-Seifer, M. J. Mataric. "Defining Socially Assistive Robotics". in Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics, 2005, pp. 465 - 468.

[9] C. Burgar, P. Lum, P. Shor, and H. Van der Loos. "Development of robots for rehabilitation therapy: The palo alto va/standford experience." in *Journal of Rehabilitation Research and Development*, vol. 37, no. 6, Nov. - Dec. 2002, pp. 663–673.

[10] S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L. Shen Yu. "PAMM - a robotic aid to the elderly for mobility assistance and monitoring" in *IEEE International Conference on Robotics and Automation*, San Francisco, volume 1, April 2000, pp. 570–576.

[11] W. Harwin, A. Ginige, and R. Jackson. "A robot workstation for use in education of the physically handicapped." in *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 2, Feb 1988, pp. 127–131.

[12] L. Kahn, M. Verbuch, Z. Rymer, and D. Reinkensmeyer. "Comparison of robot-assisted reaching to free reaching in promoting recovery from chronic stroke." in *Proceedings of the International Conference on Rehabilitation Robotics*, Evry, France, April 2001, pp. 39–44.

[13] R. Mahoney, H. Can der Loos, P. Lum, and C. Burgar. "Robotic stroke therapy assistant." in *Robotica*, 2003, vol. 21, pp. 33–44.

[14] P. Aigner and B. McCarragher. "Shared control framework applied to a robotic aid for the blind." in *Control Systems Magazine, IEEE*, vol. 19, no. 2, April 1999, pp. 40 - 46.

[15] J. Glover, D. Holstius, M. Manojlovich, K. Montgomery, A. Powers, J. Wu, S. Kiesler, J. Matthews, and S. Thrun. "A robotically-augmented walker for older adults." Technical Report CMU-CS-03-170, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, 2003.

[16] R. Simpson and S. Levine. "Development and evaluation of voice control for a smart wheelchair." in *Proceedings of the Rehabilitation Engineering Society of North America Annual Conference*, Pittsburgh, PA, June 1997, pp. 417 - 419.

[17] H. Yanco. "Evaluating the performance of assistive robotic systems." in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, Gaithersburg, MD, August 2002.

[18] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsh, D. Magaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun. "Towards personal service robots for the elderly." in *Proceedings of the Workshop on Interactive Robots and Entertainment*, Pittsburgh, PA, April-May 2000.

[19] C. Plaisant, A. Druin, C. Lathan, K. Dakhane, K. Edwards, J. Vice, and J. Montemayor. "A storytelling robot for pediatric rehabilitation." in *Proceedings of the fourth international ACM conference on Assistive technologies*, Arlington, VA, 2000, pp. 50 - 55.

[20] K. Wada, T. Shibata, T. Saito, and K. Tanie. "Analysis of factors that bring mental effects to elderly people in robot assisted activity." in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 2, Lausanne, Switzerland, October 2002, pp. 1152 - 1157.

[21] B. Graf, M. Hans, J. Kubacki, and R. Schraft. "Robotic home assistant care-o-bot II." in *Proceedings of the Joint EMBS/BMES Confrerence*, vol. 3, Houston, TX, October 2002, pp. 2343 - 2344.

[22] A. Giminez, C. Balaguer, S. M. Sabatini, and V. Genovese. "The MATS robotic system to assist disabled people in their home environments." in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 3, Las Vegas, Nevada, October 2003, pp. 2612 - 2617.

[23] K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay. "Intelligent robotic systems in service of the disabled." in *Proceedings of IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 1, March 1995, pp. 14 - 21.

[24] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro. "Person identification and interaction of social robots by using wireless tags." in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003)*, Las Vegas, NV, October 2003, pp. 1657 - 1664.

[25] T. Fong, I. Nourbakhsh, and K. Dautenhahn. "A survey of socially interactive robots." in *Robotics and Autonomous Systems*, vol. 42, no. 3 -4, 2003, pp. 143 - 166.

[26] A. Tapus, M. J. Mataric, B. Scassellatti. "Socially Assistive Robotics", in *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, 2007, pp. 35-42.

[27] K. Dautenhahn, K. Eder, M. Fisher, F. Amirabdollahian, T. Pipe, and C. Dixon. "Case for Support: Trustworthy Robotic Assistants", Submitted for funding to the EPSRC. March 2012.

[28] T. Ono, and M. Imai. "Reading a Robot's Mind: A Model of Utterance Understanding based on the Theory of Mind Mechanism", in *Advanced Robotics*, vol. 14, no. 4, 2000, pp. 311 - 326.

[29] W. P. Chan, C. A. C. Parker, H. F. Machiel Van der Loos, and E. A. Croft. "Grip Forces and Load Forces in Handovers: Implications for Designing Human-Robot Handover Controllers" in *HRI '12 Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 9 - 16.

[30] R. R. Murphy, and D. D. Woods. "Beyond Asimov: The Three Laws of Responsible Robotics", in *IEEE Intelligent Systems*, vol. 24, issue 4, pp. 14 - 20, July - Aug. 2009.

[31] D. Woods, and E. Hollnagel. "Joint Cognitive Systems: Patterns in Cognitive Systems Engineering". Taylor and Francis, 2006.

[32] D. Woods, "Conflicts between Learning and Accountability in Patient Safety". in *DePaul Law Rev.*, vol. 54, 2005, pp. 485 - 502.

[33] S. W. A. Dekker. "Just Culture: Balancing Safety and Accountability". Ashgate, 2008.

[34] D. D. Woods and N. Starter. "Learning from Automation Surprises and 'Going Sour' Accidents". in *Cognitive Engineering in the Aviation Domain*, N. B. Sarter and R. Amalberti, eds., Nat'l Aeronautics and Space Administration, 1998.

[35] Robot-Era. "Robot-Era. Implementation and integration of advanced Robotic systems and intelligent Environments in real scenarios for the ageing population". Internet: http://www.robot-era.eu/robotera/index.php, April, 2012 [April 28, 2012].

[36] Robot-Era. "Robot-Era. Implementation and integration of advanced Robotic systems and intelligent Environments in real scenarios for the ageing population". Internet: http://www.robot-era.eu/robotera/index.php?pagina=pagine_personalizzate&blocco=71&id=113, April, 2012 [April 28, 2012].

[37] Robot-Era. "Robot-Era. Implementation and integration of advanced Robotic systems and intelligent Environments in real scenarios for the ageing population". Internet: http://www.chrisfp7.eu/index.html, 2012 [April 28, 2012].

[38] R. J. Firby. "Adaptive Execution in Complex Dynamic Worlds" PhD thesis, Yale University, USA, May 1989.

[39] Rodney A. Brooks. (1986, March). "A Robust Layered Control System for a Mobile Robot". *IEEE Journal on Robotics and Automation*, 2(1), pp. 14 - 23. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1087032 [April 28, 2012].

[40] E. Gat, R. P. Bonnasso, R. Murphy, and A. Press. "On Three-Layer Architectures". in *Artificial Intelligence and Mobile Robotics*, AAAI Press, 1997, pp. 195 - 210.

[41] M. J. Wooldridge. *Reasoning about Rational Agents*. MIT Press, 2000.

[42] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.

[43] A. S. Rao, and M. P. Georgeff. "An Abstract Architecture for Rational Agents". in *Proc. Int. Conf. Knowledge Representation and Reasoning (KR & R)*, 1992, pp. 439 - 449.

[44] M. P. Georgeff, and F. F. Ingrand. "Decision-Making in an Embedded Reasoning System." in *Proc. International Joint Conference on AI*, 1989, pp. 972 - 978.

[45] H. Chockler, O. Kupferman, and M. Vardi, "Coverage Metrics for Formal Verification" in *Lecture Notes in Computer Science: Correct Hardware Design and Verification Methods*, D. Geist and E. Tronci, Eds. Berlin: Springer, vol. 2860, 2003, pp. 111-125.

[46] E. M. Clarke, "Model Checking" in *Lecture Notes in Computer Science: Foundations of Software Technology and Theoretical Computer Science*, vol. 1346, 1997, pp. 54 - 56.

[47] G. J. Holzmann. *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.

[48] M. Huth, and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd ed. Cambridge University Press, 2004.

[49] T. Ball, and S. K. Rajamani. "The SLAM Toolkit". in *Lecture Notes in Computer Science: Computer Aided Verification*, vol. 2012, 2001, pp. 260 - 264.

[50] S. Berezin, E. M. Clarke, A. Biere, and Y. Zhu. (2002). "Verification of Out-of-Order Processor Designs Using Model Checking and a Light-Weight Completion Function". *Formal Methods in System Design*, 20(2), pp. 159 - 186.

[51] R. H. Bordini, M. Fisher, W. Visser, and M. Wooldrige. (2004, Sept. - Oct.). "Model Checking Rational Agents". *IEEE Intelligent Systems*, 19(5), pp. 46 - 52.

[52] F. Raimondi, and A. Lomuscio. (2007, June). "Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams". *Journal of Applied Logic*, 5(2), pp. 235 - 251.

[53] L. A. Dennis, M. Fisher, M. Webster, and R. H. Bordini. (2012). "Model Checking Agent Programming Languages". *Automated Software Engineering*, 19(1), pp. 5 - 63.

[54] L. A. Dennis, B. Farwer, R. H. Bordini, M. Fisher, and M. Wooldridge. "A Common Semantic Basis for BDI Languages". in *ProMAS '07 Proceedings of the 5th international conference on Programming multi-agent systems*, vol. 4098, 2007, pp. 124 - 139.

[55] B. Wile, J. C. Goss, and W. Roesner. *Comprehensive Functional Verification*. San Francisco: Morgan Kaufmann Publishers, 2005.

[56] H. Foster, A. Krolnik, and D. Lacey. *Assertion-based Design*, 2nd edition, Massachusetts: Springer, 2004.

[57] G. Metta, L. Natale, S. Pathak, L. Pulina, and A. Tacchella. "Safe and Effective Learning: a Case Study" in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4809 - 4814.

[58] S. Bensalem, M. Gallien, F. Ingrand, I. Kahloul, and T. H. Nguyen. (2009, March). "Designing Autonomous Robots: Toward a More Dependable Software Architecture". *IEEE Robotics and Automation Magazine*, 16(1), pp. 67 - 77.

[59] A. Basu, M. Gallien, C. Lesire, T. H. Nguyen, S. Bensalem, F. Ingrand, and J. Sifakis. "Incremental Component-Based Construction and Verification of a Robotic System" in *Proc. 18th European Conference on Artificial Intelligence*, 2008, pp. 631 - 635.

[60] G. Pappas, and H. Kress-Gazit (eds.). *ICRA Workshop on Formal Methods in Robotics and Automation*, 2009.

[61] D. F. Gordon. "APT Agents: Agents That Are Adaptive, Predictable, and Timely". in *Lecture Notes in Computer Science: Formal Approaches to Agent-Based Systems*, vol. 1871, 2001, pp. 278 - 293.

[62] M. Gudemann, F. Ortmeier, and W. Reif. "Safety and Dependability Analysis of Self-Adaptive Systems". in *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, 2006, pp. 177 - 184.

[63] M. L. Bolton, and E. J. Bass. "Building a Formal Model of a Human-interactive System: Insights into the Integration of Formal Methods and Human Factors Engineering". in *The First NASA Formal Methods Symposium*, 2009, pp. 6 - 15.

[64] S. D. Stoller, E. Bartocci, J. Seyster, R. Grosu, K. Havelund, S. A. Smolka, and E. Zadok. "Runtime Verification with State Estimation" in *Proceedings of the 2nd International Conference on Runtime Verification*, 2011, San Francisco.

[65] L. R. Rabiner. (1989, Feb.). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" in *Proceedings of the IEEE*, 77(2), pp. 257 - 286.

[66] A. P. Dempster, N. M. Laird, and D. B. Rubin. (1977) "Maximum Likelihood from Incomplete Data via the EM Algorithm". *J. Roy. Stat. Soc.* 39(1), pp. 1 - 38.

[67] J. A. Bilmes. "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models". International Computer Science Institute, 1998.

[68] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. (1983, April). "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition". *Bell Syst. Tech*, 62(4), pp. 1035 - 1074.

[69] J. K. Baker. (1975, Feb.). "The dragon system - An overview". *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-23(1), pp. 24- 29.

[70] L. E. Baum, and G. R. Sell. (1968). "Growth functions for transformations on manifolds". *Pac. J. Math. Stat.*, 27(2), pp. 211 - 227.

[71] L. A. Liporance. (1982). "Maximum likelihood estimation for multivariate observations of Markov sources". *IEEE Trans. Informat. Theory*, IT-28(5), pp. 729 - 734.

[72] A. B. Poritz. "Linear predictive hidden Markov models and the speech signal" in *Proc. ICASSP '82*, Paris, France, 1982, pp. 1291 - 1294.

[73] B. H. Juang, and L. R. Rabiner. (1985, Dec.). "Mixture autoregressive hidden Markov models for speech signals". *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-33(6), pp. 1404 - 1413.

[74] L. R. Bahl, F. Jelinek, and R. L. Mercer. (1983). "A maximum likelihood approach to continuous speech recognition". *IEEE Trans. Pattern Anal .Machine Intell.*, PAMI-5, pp. 179 - 190.

[75] M. J. Russell, and R. K. Moore. "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition" in *Proc. ICASSP '85*, Tampa, Fl., 1985, pp. 5 - 8.

[76] S. E. Levinson. (1986, Mar.). "Continuously variable duration hidden Markov models for automatic speech recognition". *Computer, Speech, and Language*, 1(1), pp. 29 - 45.

[77] S. Shibata, B. M. Sahbi, K. Tanaka, and A. Shimizu. "An Analysis of The Process of Handing Over an Object and Its Application to Robot Motion" in *Systems, Man, and Cybernetics, IEEE International Conference on Computational Cybernetics and Simulation*, vol. 1, 1997, pp. 64 - 69.

[78] M. Huber, A. Knoll, T. Brandt, and S. Glasauer. (2009, May). "Handing Over a Cube - Spatial Features of Physical Joint-Action". *Annals of the New York Academy of Science*, vol. 1164, Basic and Clinical Aspects of Vertigo and Dizziness, pp. 380 - 382.

[79] S. Shibata, B.M. Sahbi, K. Tanaka, and A. Shimizu. (1997, Apr.). “An Analysis of The Process of Handing Over An Object and Its Application to Robot Motions”. *SMC*, vol. 1, pp. 64 - 69.

[80] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. “Human-Robot Interaction in Handing-Over Tasks” in *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*, 2008, pp. 107 - 112.

[81] P. Basili, M. Huber, T. Brandt, S. Hirche, and S. Glasauer. (2009). “Investigating Human-Human Approach and Hand-Over”. *Human Centered Robot Systems*, vol. 6, pp. 151 - 160.

[82] C. Becchio, L. Sartori, and U. Castiello. (2010, Jun.). “Toward You: The Social Side of Actions”. *Current Directions in Psychological Science*, vol. 19, pp. 183 - 188.

[83] M. Cakmak, S.S. Srinivasa, M.K. Lee, S. Kiesler, and J. Forlizzi. “Using Spatial and Temporal Contrast for Fluent Robot-Human Hand-overs.” in *6th ACM/IEEE International Conference on Human-Robot Interaction*, 2011.

[84] A. Edsinger and C.C. Kemp. “Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another” in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, 2007, pp. 1167 - 1172.

[85] F. Shic, W. Jones, A. Klin, and B. Scassellati. "Swimming in the Underlying Stream: Computational Models of Gaze in a Comparative Behavioral Analysis of Autism" in *Cognitive Science*, Vancouver, 2006, pp. 780 - 785.

[86] S. J. Luck, S. A. Hillyard, M. Mouloua, and H. L. Hawkins. (1996). "Mechanisms of visual-spatial attention: Resource allocation or uncertainty reduction". *Journal Exp. Psych.: Human Perception and Performance*, vol. 22, pp. 725-737.

[87] M. Hayhoe, and D. Ballard. (2005). "Eye Movements in natural behavior". *Trends in Cog. Sci.* 9(4), pp. 188 - 194.

[88] S. Baron - Cohen. *Mind Blindness: An Essay on Autism and Theory of Mind*. Cambridge: MIT Press, 1995.

[89] A. M. Leslie. (1994). "ToMM, ToBY, and Agency: Core architecture and domain specificity." in *Mapping the Mind: Domain Specificity in Cognition and Culture*, L.A. Hirschfeld and S. A. Gelman (Eds.), Cambridge University Press: Cambridge, pp. 119 - 148.

[90] Phidgets. "Products for USB Sensing and Control - PhidgetInterfaceKit 8/8/8 ". Internet: http://www.phidgets.com/products.php?category=0&product_id=1018_2, 2011 [May 2, 2012].

[91] M. Mori. "The Uncanny Valley", in *Energy*, vol. 7, no. 4, 1970, pp. 33-35.

[92] R. S. Nagayama, "The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces" in *Presence*, vol. 16, no. 4, 2007, pp. 337-351.

[93] S. Sutton, R. Cole, J. D. Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, K. Shobaki, P. Hosom, A. Kain, Johan, J. Wouters, D. Massaro, and M. Cohen. “Universal speech tools: The cslu toolkit,” IN *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING ICSLP*, vol. 7, 1998, pp. 3221-3224.

[94] C. X. Zhai. "A Brief Note on the Hidden Markov Models (HMM)", Department of Computer Science , University of Illinois at Urbana-Champaign, March 16, 2003.

[95] F. Paas, A. Renkl, and J. Sweller. (2003). "Cognitive Load Theory and Instructional Design: Recent Developments". *Educational Psychologist*, 38(1), pp. 1 - 4.

7 Appendix

7.1 Ethical Approval

University of the West of England
BRISTOL

FACULTY OF ENVIRONMENT & TECHNOLOGY

FACULTY RESEARCH ETHICS COMMITTEE

Your ethics application was considered under Chair's Action.

Committee Reference No.	FETREC11-12/08
Name of Applicant	Elena Corina Grigore
Home Faculty	FET
Title of Proposal	I Robot, I Learn
Outcome	The Chair is content to **approve** the application subject to the following: • You notify the Faculty Research Ethics sub Committee in advance if you wish to make significant amendments to the original application; • You notify the Faculty Research Ethics sub Committee if you terminate your research earlier than planned. • Supervisor has not checked and signed on page 2. This can be an email signature. • The consent form needs a field for the participant's name to be clearly written as well as for their signature; otherwise there is a risk of non-identification (some people have illegible signatures.)

Date 17/10/11

Signed

Mark Palmer

Dr. Mark Palmer, Chair, Faculty Research Ethics Committee

7.2 Experiments Information for Candidates

Bristol Robotics Laboratory
University of Bristol and University of the West of England
T Block, Frenchay Campus
Bristol UK
BS34 8QZ

Bristol, May 2012

Information Sheet Experiment:
I Robot, I Think

In this experiment, our robot (BERT2) is acting as a drink-serving robot. Its main task is to hand you over the drink. It is then able to take the decision of releasing the cup based on a state estimation of the interaction between you and the robot and on other cues picked up from you.

The experimental setting is as follows. You will be seated in a chair in front of the robot. This is the position you should maintain during the entire experiment. This does not mean that you have to sit completely still, but you must not move from your chair. During the entire experiment, your task is simply to answer the robot's questions to order a drink and to try to take the drink from BERT2.

During the experiments, the researcher will sometimes ask you to say out loud the sequence of numbers obtained by starting with 4 and keeping on adding 7 each time. This is in order to simulate what would happen in the context of the user being engaged in a different activity while trying to get a cup from a robot.

Example of dialog:

• Robot: "Would you like a drink?"

• Robot: "Please get the drink."

• etc ...

After the experiment you will have the opportunity to ask the researchers about their work.

7.3 Experiments Consent Form

Bristol Robotics Laboratory
University of Bristol and University of the West of England
T Block, Frenchay Campus
Bristol UK
BS34 8QZ

Bristol, May 2012

Consent Form
Experiment: I Robot, I Think

I have read and understood the information sheet and this consent form. I have had an opportunity to ask questions about my participation.

I understand that I am under no obligation to take part in this study.

I understand that I have the right to withdraw from this study at any stage without giving any reason.

I agree that recording of interaction data will be stored anonymously and may be used for future analysis by other researchers associated with Bristol Robotics Laboratory.

I agree to participate in this study.

Name of participant: ______________________________________

Signature of participant: ______________________________________

Signature of researcher: ______________________________________

Date: ______________________________________

I agree that my interaction with the robot may be filmed.

Signature of participant: ______________________________________

Signature of researcher: ______________________________________

Date: ______________________________________

Contact details of the researcher:

Elena Corina Grigore
Undergraduate MEng Student, University of Bristol
Computer Science Department

Phone: +44 (0)7899675415
eg9542@bris.ac.uk

7.4 Project Poster

Bristol Robotics Laboratory

University of the West of England BRISTOL

University of BRISTOL

I Robot, I Think

Elena Corina Grigore
Dept. of Computer Science
University of Bristol

Kerstin Eder
Dept. of Computer Science
University of Bristol

Ute Leonards
School of Experimental Psychology
University of Bristol

Objective

- Within the scenario of a humanoid robot serving a drink to a human user, make the robot take the decision of when it is safe to release the drink based on the classification of the state the system is in (e.g. the robot is holding the cup, the human is grabbing the cup, etc.)
- Investigate parameters needed to ensure that the hand-over is safe from a human behavioural perspective: the actions and intentions of the user are taken into account in order to account for the effect realistic variations in the environment might have on the user's behaviour.

Context and Motivations

Human-assistive robots can become an important part of our future if we manage to design useful and safe machines that will adequately interact with humans in helping them to achieve various tasks, examples are robot waiters, personal care robots helping patients in hospitals during recovery.

Human Intentions

- Collaboration with an experimental psychologist has highlighted the importance of the following sequence of actions the robot should expect from a user: the user looks at the cup, the user looks away, the user touches the cup.
- Although there are variations in the amounts of time each of the above actions takes, the robot will not consider it safe to release the cup unless this sequence is executed. Not performing one of these actions means something has changed in the environment and the user is either not able to or does not want to get the cup (e.g. someone calls the user).

Implementation

- System state classifier is implemented by using a Hidden Markov Model and taking in the motor current values of the robot's hand as observations.
- The decision of releasing the cup is taken by gathering all the information from the different running modules: the gaze module which informs the robot if the user looked at the cup and the module which informs the robot if the user actually touched the cup.

Experiments

- A series of experiments was conducted to compare the two implementations and find out if modelling user intentions has any effects on the safety aspect.
- Different scenarios were tested in order to ensure reliability.

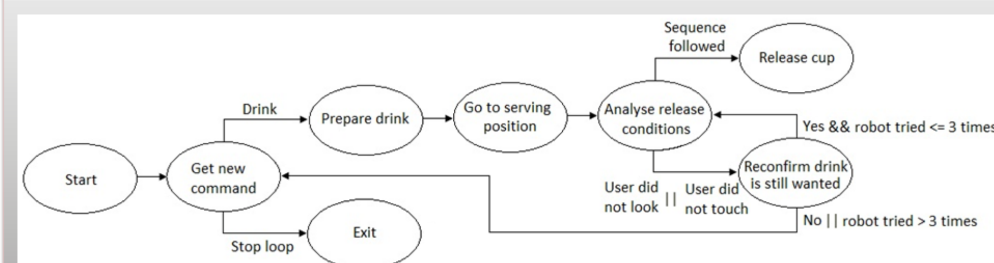
Results

- Experiments proved that the extended system which modelled the user's intentions is safer, and more reliable than the basic HMM system.
- The "theory of mind" concept proved to be an important part of the model.

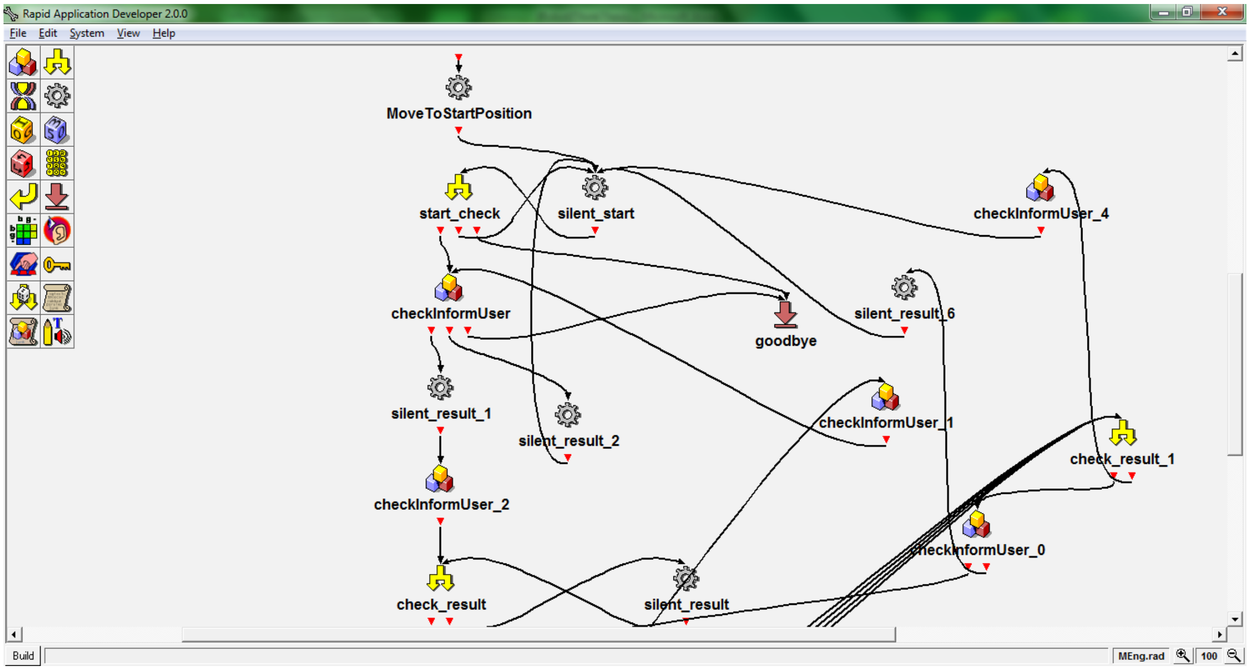
Conclusions and Future Work

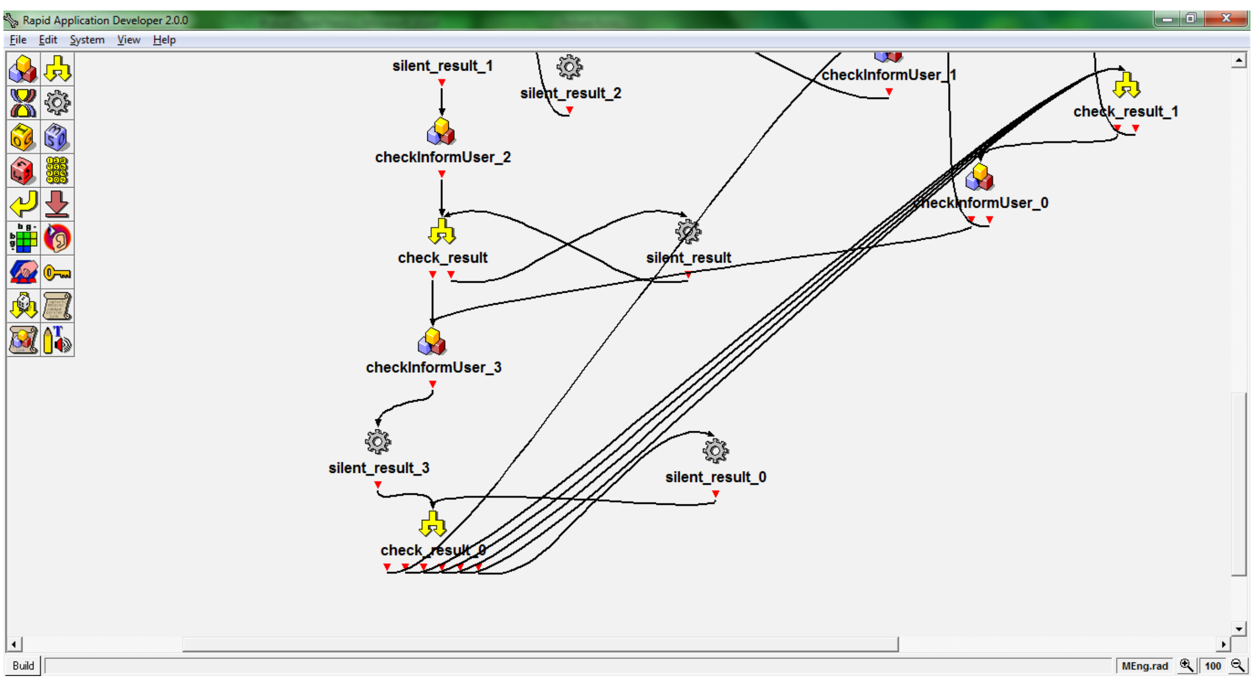
- Modelling user intentions provides the system with important information about what the next expected move of the human is.
- Future research direction include developing methods for formal verification and validation, verification at runtime, and further investigations into the "theory of mind" concept.

State Machine from Robot's viewpoint



7.5 RAD Dialogue for Verbal Communication





7.6 Implementation Code Sample - Checking that the user is looking at the cup

```
cout<<"User touched the cup.\n";
//get info from gaze module
bottleIGaze.clear();
if(!readOnce(bottleIGaze,portI6))
	cout<<"Problem with receiving input from gaze module."<<endl;
intervalStart = bottleIGaze.get(1).asInt();
intervalEnd = bottleIGaze.get(2).asInt();
timeNow = bottleIGaze.get(3).asInt();
difference = bottleIGaze.get(4).asInt();
cout<<bottleIGaze.get(0).asInt()<<" "<<intervalStart<<" "<<intervalEnd<<" "<<timeNow<<"
"<<difference<<endl;

if((intervalEnd < intervalStart && difference > GAZE_THRESH_MIN) || (intervalEnd >=
intervalStart && difference < GAZE_THRESH_MAX))
{
	cout<<"Inside gaze check.\n";
	cout<<" time difference: "<<timeNow - intervalStart<<"\n";
	//check again if user is still touching the cup
	bottleIHand.clear();
	if(!readOnce(bottleIHand,portI3))
		cout<<"Problem with receiving input from hand module."<<endl;
	if(!bottleIHand.get(0).asInt())
	{
		//release cup
		moveHand(0); //open grip
		Time::delay(1);
		//send to voice system "YesTouchYesLook"
		bottleOVoice.clear();
		bottleOVoice.add("YesTouchYesLook");
		if(!writeOnce(bottleOVoice,portO1))
			cout<<"Problem with sending output."<<endl;
			cout<<"should have sent it"<<endl;
			Time::delay(2);
			//getHandArmReadyForServe();
			state = "StartAgain";
	}
	else //user looked at the cup, but is not touching it anymore
	{
		state = "StartAgainInnerLoop";
		//send to voice system "NoTouchYesLook"
		bottleOVoice.clear();
		bottleOVoice.add("NoTouchYesLook");
		if(!writeOnce(bottleOVoice,portO1))
			cout<<"Problem with sending output."<<endl;
		cout<<"should have sent it"<<endl;
	}
}

....
```