Towards Safe Human-Robot Interaction*

Elena Corina Grigore[1], Kerstin Eder[1,2], Alexander Lenz[2], Sergey Skachek[2],
Anthony G. Pipe[2], Chris Melhuish[2]

[1] Computer Science Department, University of Bristol, Bristol BS8 1UB
{`Kerstin.Eder`}`@bristol.ac.uk`
[2] Bristol Robotics Laboratory, Bristol Business Park, Frenchay, Bristol BS16 1QD
{`Alex.Lenz,Sergey.Skachek,Tony.Pipe,Chris.Melhuish`}`@brl.ac.uk`

Abstract. The development of human-assistive robots challenges engineering and introduces new ethical and legal issues. One fundamental concern is whether human-assistive robots can be trusted. Essential components of trustworthiness are usefulness and safety; both have to be demonstrated before such robots could stand a chance of passing product certification. This paper describes the setup of an environment to investigate safety and liveness aspects in the context of human-robot interaction. We present first insights into setting up and testing a human-robot interaction system in which the role of the robot is that of serving drinks to a human. More specifically, we use this system to investigate when the right time is for the robot to release the drink such that the action is both safe and useful. We briefly outline follow-on research that uses the safety and liveness properties of this scenario as specification.

1 Introduction

Human-assistive robots are machines designed to improve our quality of life by helping us to achieve tasks, e.g. a personal care robot might help us during accident recovery. Such robots will perform physical tasks within our personal space, including shared manipulation of objects and even direct contact. The capability to dynamically adapt to different situations is a prerequisite for these robots to be genuinely useful in practice. To be effective they may also need to be powerful. The combination of both these capabilities makes them potentially dangerous and capable of inflicting significant harm to humans or damage to surrounding objects.

The development of human-assistive robots challenges engineering and introduces new ethical and legal issues that need to be addressed to unlock the clear potential for the huge social and financial benefit that is expected to result from large-scale commercial exploitation. One fundamental concern is whether human-assistive robots can be trusted. Essential components of trustworthiness

---

* This work was supported in part by the European Union under the project "Cooperative Human Robot Interaction Systems (CHRIS)"; grant agreement No: 215805.

are usefulness[3] and safety[4]. Demonstrable trustworthiness will be a prerequisite for such robots to pass product certification. This, in turn, will be a fundamental requirement for any large-scale commercialisation of robots that operate in close proximity to humans. Development of a method for undertaking such certification is currently a wide open research question.

This paper describes the setup of an environment to investigate safety and liveness aspects in the context of close Human-Robot Interaction (HRI), i.e. HRI demanding co-location in the same physical space and collaborative interaction with physical objects. We present first insights into setting up and testing a HRI system in which the role of the robot is that of serving drinks to a human. The setting includes a humanoid robot head capable of gaze-tracking, a voice system that allows the user to interact directly with the robot, and a robot hand which grabs a cup from an initial position and takes it to a serving position; the robot must decide whether to release the cup or not.

To ensure safety and liveness of this hand-over we propose safety and liveness properties. We have conducted initial experiments to find safe and useful settings for the control parameters in the system, e.g. the amount of pressure to be applied for the robot to release the drink. The tests performed on this particular system, dealing with handing over and releasing a drink, can be a helpful exemplar for other HRI tasks such as handling of objects (e.g. robot teaching/playing with a child), or even more direct interaction such as shaking hands (this is where the pressure sensors would come in, as well as identifying the position of the human's hand). We have identified research questions for challenging follow-on research projects that introduce machine learning to enable the robot to adapt its behaviour to different situations.

2 Related Work

One corner stone for successful verification is a well defined specification. This is particularly important for adaptive systems because the system changes over time adapting to different situations. Verification is the process used to demonstrate the correctness of a design with respect to the specification. Formal methods have been used to establish the correctness of adaptive systems either by verification or by systematic design. An example is the application of model checking [3] to prove that a specification in the form of a set of safety properties is satisfied by a multi-agent control system that adapts by learning from experience [8]. Other work has focused on ensuring safety by construction [2]. Systems such as the LAAS Architecture for Autonomous System, which is based on the BIP (Behaviors Interactions Priorities) framework, support a systematic component-based design, construction and verification approach which can

---

[3] Aka liveness, i.e. robots don't fail to do things they are supposed to do - cf. the definition of a *liveness property* "something good must eventually happen" [7].

[4] Robots operate within their specification and within safe limits - cf. the definition of a *safety property* "something bad does never happen" [7].

enforce online safety properties [1]. The importance of formal methods has recently been recognized at the ICRA workshop "Formal Methods in Robotics and Automation" [9]. As systems become more and more adaptive, however, the verification challenge increases and new scalable solutions need to be found.

The work presented here prepares the ground for research into finding a compromise between adaptability and verifiability, so as to identify those situations where the emphasis should be put more on one, or the other. We introduce a set of safety and liveness properties acting as specification. Even an apparently simple scenario, such as offering a drink to a person whilst being simultaneously safe and useful, is a challenge because every user is different and will react in a slightly different way to the robot. As the choices increase, and the robot's behavior becomes more adaptive, the more difficult it is to verify it. On the other hand, offering only a small fixed set of behaviors would seriously limit the usefulness of the robot. Clearly, the specification of safety and liveness properties plays a key role in making advances towards verifying adaptive systems.

3 System and Human-Robot Interface

The context of the project is a "drink serving" robot, that interacts with a human to pass over a drink. The user is offered two options, water or coffee, and, dependent on this choice, the robot will need to satisfy different safety constraints. Water is the less dangerous choice, while coffee is more dangerous, because it can stain or burn the user if spilled. The number of choices was limited to two for this initial explorative study.

Our implementation included various components. We used the humanoid head from [5] as shown in Figure 1. A hand/wrist device, monitored by a VICON motion capture suite, was used for tracking the position of the user's hand. A simple robot arm was used to move the cup from a set initial position to the serving position. Finally, a voice system allows for bidirectional vocalization.

For the basic scenario, the robot would use all these cues to identify whether the user is looking at the cup and holding it, so that it can be released in a safe manner. The robot can, however, never be entirely sure if the user is holding the cup or not just by using the position of the hand/wrist. Thus, we advanced the scenario by adding pressure sensors to the robot's hand. Although pressure sensors can introduce new problems themselves, depending on how each person applies pressure to the cup, the introduction of pressure sensors was expected to provide a safer more reliable solution. We compare the two scenarios in Section 7.

BERT2 [6] uses the VICON motion capture (MoCap) system to detect and localise interaction objects and the human's body parts in 3D space. Using a MoCap system as the main vision facility for a humanoid robot may seem far fetched (as it is an external system that cannot be fitted to the robot). However, this allows us to divert away from the machine vision challenges typically encountered when using robot mounted vision systems. Instead, we can directly focus on the subject of HRI. The VICON software stores information about body and object marker topology.

Fig. 1. BERT2's expressive head. An embedded LCD screen generates gestures. Animated eyes communicate the direction of attention. The stereo vision system faceLAB from Seeing Machines^{TM} is mounted on the side of the head for head and gaze tracking. A webcam is mounted centrally in the forehead for a wide angle view of the scene.

4 Software Architecture

An overview of the software architecture is presented in Figure 2. The following briefly describes each module and inter-module interactions.

The *Main Module* communicates the state of the system at essential points in time to other modules and gathers results about the individual components at that moment. It triggers other modules based on its synchronization with the Voice System. For example, after the Voice System tells the user *"Please hold the cup."* the Main Module sends triggers to the Gaze Tracking and the Hand Wrist Modules to start checking whether or not the user is looking at and holding the cup. When the two modules have the results, they send them to the Main Module, which in turn controls the Voice System.

The *Voice System* is utilized to ask the user for the drink of their choice, and for the robot to utter a small set of instructive sentences. This spoken language interface is based on the Festival speech synthesis and Sphinx-II speech recognition systems. The *EgoSphere* acts as object position and orientation storage. Software modules in the architecture are interconnected using *YARP* [4], an open source library that provides an intercommunication layer to allow data exchange between processes running on different machines. Further information about the above four components can be found in [5] and [6].

The *Gaze Tracking Module* estimates the human's focus of attention using faceLAB^{TM}, an integrated head and gaze tracking system [5].

When triggered, the *Hand Wrist Module* starts to check, within a three second interval, the position of the user's hand. The positions of both the cup and the hand/wrist are always up-to-date because the Object Provider module runs in the background and constantly updates the EgoSphere. The module works by verifying if the user's hand/wrist is within the region defined by the circle C(O(a,b),r), where the centre O of the circle is the centre of the cup (a and b are the coordinates of this point) and the radius r is the distance between the centre of the cup and the hand/wrist. This radius obviously changes depending on the user's hand size and thus would require calibration to ensure that, if the radius is set too large, the cup will not be released when the user is not actually

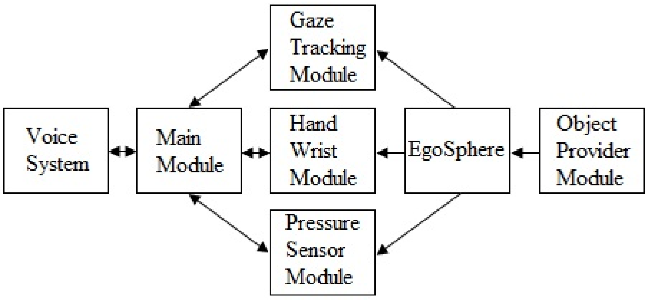

Fig. 2. Software Architecture. Arrows indicate information flow via YARP.

holding it. Alongside this circle region, the algorithm also checks whether the position of the user's hand is at the same height as the cup (within a threshold of measurement error). If, within the time interval, these constraints are satisfied relative to a determined threshold, the module will return a *true* value; otherwise it will return *false*.

When triggered, the *Pressure Sensor Module* first computes the value indicated by the sensor at that moment, averaging over 10 samples. At this point, the robot arm is holding the cup in the serving position and this is the ideal moment for the module to acquire the control value. If, over the next three seconds, the value acquired by the pressure sensor varies by at least 10%-20%, the module will return the *true* value; otherwise it will return *false*. The percentage by which the value should vary depends on how the user holds the cup and this is investigated in Section 7.

5 How the system works

The core of the algorithm checks for the main cues (hand/wrist position, gaze and, optionally, applied pressure), regardless of what choice the user has made. It then checks, depending on the choice, certain pre- or post-conditions. The conditions are checked within a three seconds time interval, during which the user needs to satisfy the requirements, e.g. looking at the cup, holding it, or applying enough pressure. Figure 3 shows the state diagram of the entire system; it is further explained below.

The basic option is considered to be *Water* and, if the user makes this choice, the system will enter the *Condition* state after it exits the *Choice* state. If all the conditions are satisfied, i.e. the user is simultaneously looking at, holding and applying pressure to the cup, the system will enter the *Release Cup* state, then wait for a period of 25 seconds, after which it will enter the *Another Cup* state. The algorithm loops while the user answers *yes* or *no*. When the user's answer is *exit*, the algorithm will stop. If at least one of the conditions is not satisfied, the algorithm will loop back to the *Condition* state until all conditions are satisfied, looping a maximum number of three times. If the algorithm has looped three

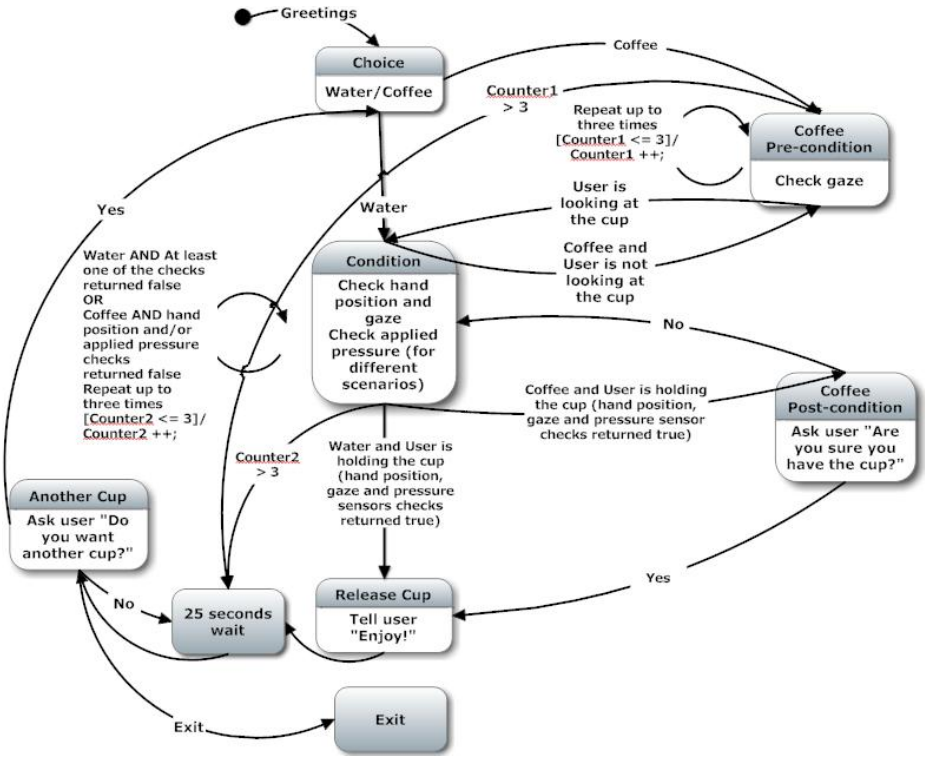

Fig. 3. State Diagram of the entire system.

times with no satisfactory results, the system will not release the cup; instead it will retract and enter the *25 seconds wait* state.

The alternative option, *Coffee*, will cause the system to enter the *Coffee Pre-condition* state. Upon satisfaction of the pre-condition, i.e. the user is mostly looking at the cup in the required time interval, the system will enter the core *Condition* state. The difference in behavior when choosing coffee is that the algorithm will loop back to the *Condition* state if the user is not holding the cup or not applying enough pressure, but continues to look at the cup (i.e. maintains the pre-condition). This will cause the robot to repeat to the user *"Please hold the cup."* If, however, the gaze check pre-condition is falsified in the *Condition* state, then the algorithm will loop back to the *Pre-condition* state and the robot will tell the user *"Please pay attention to the cup."*

This makes the robot pro-active in that if the pre-condition is subsequently falsified, meaning the user no longer looks at the cup, the robot explicitly asks the user to do so. The fact that the user then looks at the cup will have a stronger meaning than if the robot had not requested this.

When the pre-condition and the core conditions are met, the system will enter the *Coffee Post-condition* state. It asks the user: *"Are you sure you have* *the cup?"* This is a safety measure for the coffee option. It ensures the robot receives verbal confirmation that the human is ready to take the cup. If this post-condition is met, the system will behave in the same way as described above for water. For the *Coffee Pre-condition* state, the same maximum number of loops applies before the algorithm retracts and enters the *25 seconds wait* state without releasing the cup.

6 Trustworthiness

Humans need to trust a robot to be comfortable using it. Trustworthiness is delicate territory with fuzzy boundaries in practice. We analyze a robot's trustworthiness by analyzing its safety and liveness properties. These properties are key to specifying what constitutes safe and useful behaviour and should be maintained by any behavioural adaptations the robot might learn.

6.1 Safety Properties

(S1) A cup of water is never released unless the user is looking at the cup, the hand/wrist is within the circle area, and (when using hand pressure sensors) the user is applying enough pressure on the cup. All these conditions must be fulfilled simultaneously in order for the cup to be released.

(S2) A cup of coffee is never released unless the user is initially looking at the cup (pre-condition), after which the user is required to continue looking at the cup while his/her hand/wrist is within the circle area, and the user is applying enough pressure on the cup (core condition), after which the robot receives verbal confirmation that the user is holding the cup (post-condition). If any one of these conditions is not met, the system loops back to the state where it failed. If the maximum number of times the system can loop is reached, then the cup is not released and another drink is offered, i.e.:

  (S2-a) When the choice is coffee, the robot never asks the user *"Please grab/hold the cup."* unless the user previously paid attention to the cup.

  (S2-b) When the choice is coffee, the robot never asks the user *"Are you sure you have the cup?"* unless the user previously paid attention to the cup (S2-a), the hand/wrist was within the circle area, and the user applied enough pressure on the cup (implying first property is satisfied).

(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.

6.2 Liveness Properties

(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?"*

The constraint of permitting three repeats reflects the fact that a user would, in normal circumstances, be able to perform the required steps in one or two tries. When considering external factors, such as the users' lack of attention or some external event happening that prevents the user from performing the steps, these would usually allow the user to finish the process within three tries. If something is happening that is not allowing the user to finish the process, then the user has the option of stopping and starting again. Situations in which the user is impaired, e.g. blind or deaf, can be accommodated but were not the focus of this project.

7 Experiments

This project was a short-term initial investigation into finding a working set of safety and liveness properties on which further research to determine the tradeoffs between adaptability and verifiability could be based. The experiments conducted are hence of a "pilot" nature.

Although the number of participants in the experiment was only five, each session was one hour long and covered different scenarios. This allowed us to test how familiarization with the system impacts on user performance and, most importantly, how this impacts on the safety and liveness properties of the system.

7.1 Experimental Setting

The users were seated in a chair in front of the robot and they were wearing the hand/wrist device that enables the VICON system to track the hand position. They were given an information sheet detailing the experiment, as well as a participation consent form. Before starting the experiment, the voice and gaze tracking systems were calibrated.

The users directly interacted with the robot via the Voice System. Details of an interaction depended on the user's choices. A dialogue could proceed according to this example:

- Robot: "Hello, I am BERT2. Do you feel like a drink?"
- *Human: "Yes."*
- Robot: "Water or coffee?"
- *Human: "Water."*
- Robot: "Preparing drink."
  (Pause while the robot is preparing the drink. When ready the robot hand moves upwards with the cup to the serving position.)
- Robot: "Please hold/grab the cup."
  (The human should now hold/grab the cup.)
- Robot: "Enjoy!"

(The robot releases the cup and the user can have the drink.)

The above script was for the water option. For the coffee option, the robot will also say *"Please pay attention to the cup."* and *"Are you sure you have the cup?"* as these two lines represent the pre- and post-condition, respectively as detailed in Section 5.

7.2 Description of Experiments

Each experiment consisted of three validation scenarios. The first one was without pressure sensor. The second one used the pressure sensor with a low threshold that allowed the cup to be released when the user holds the cup without having to pull it (the robot states *"Please hold the cup."*). The third scenario used the pressure sensor with a slightly higher threshold requiring the user to apply some force for the cup to be released (the robot states *"Please grab the cup."*)

The motivation for having two types of scenarios, with and without the pressure sensor, was to improve upon the first scenario by making the system safer. We expected to observe more failures in experiments without the pressure sensor.

The two pressure sensor settings are used to investigate user reactions. If the person is asked to hold the cup, we expect that less force is applied and the user's hand will remain on the cup until release. If the person is asked to grab the cup, we expect that more force is applied and the user will actively try to pull the cup from the robot's hand. Thus, we intend to investigate whether the cup hand-over is successful more often when the user just holds the cup or when the user actively has to pull it from the robot's hand.

For each validation scenario, we assigned three types of tests:

- *Standard test:* The user is asked to act as if this were a real-world situation, when trying to buy a cup of water/coffee from a robot.
- *Interruption test:* As above but the researcher calls the participant's name at the point in the experiment when the robot says *"Please pay attention to the cup."* or *"Please hold/grab the cup."*
- *Conversation test:* The user is engaged in a conversation and asked to interact with the robot at the same time.

7.3 Expected Outcomes

For the first scenario, without the pressure sensor, the robot action was expected not to be safe, i.e. the cup would be released when the user was not actually holding it or the robot would fail to release it when it should (because of the varying distance between the cup and the hand/wrist).

The second and third scenarios include the pressure sensor which was expected to make the release of the cup safer. While this eliminated the need to calibrate the system with respect to each user's hand length, it required finding an appropriate threshold for the release pressure; we investigated two settings.

Each type of test was to be performed three times and the test type order within each scenario was random. This would enable users to perform better each time and the variation of the results between users for the last sets of tests was expected to be lower than at the start.

The methods of data collection used were logging the threshold used for the distance from the centre of the cup to the hand/wrist, the thresholds for the pressure sensors[5] (the threshold for the low pressure scenario varied around 0.01 and the one for the higher pressure scenario varied around 0.2), the number of times the values were between the required thresholds and the number of system failures. The testing sessions were also video recorded.

8 Results

Experiments concentrated on evaluating our system with respect to the set of safety and liveness properties from Section 6. The first scenario focused on the distance from the centre of the cup to the hand/wrist and, thus, the threshold which needed to be determined in this situation. During testing and debug, this threshold was set to 3 cm which fitted the people testing the system. For the experiments the threshold was changed to 5 cm in order to start with a less limiting value with a view to further constrain (i.e. lower) it if the system failed because of it. This threshold actually proved to be quite robust, as it was loose enough to give the impression that the users' hand positions were within the intended circle region. Of course, this threshold would have needed to be modified in the case where the system released the cup when the user's hand was close to the cup, but not actually holding it. This was the very reason for designing the interruption type of test, which would check if the robot would release the cup in the case where the user's hand might, or might not, stay close to the cup when being called.

In preliminary testing, when the hand was intentionally held within the circle delimited by the threshold but the hand was not actually touching the cup, then the system would incorrectly release the cup because it classified this situation as the cup was being touched. This motivated the introduction of the interruption tests. When tested with subjects who were not previously exposed to the system, however, the tests revealed the following results: if the users had already moved their arms towards the cup for more than half of the distance, when called, they would touch the cup, but not look at it; if the users had not moved their arms that far towards the cup, they did not continue the movement. Thus, there was no moment, in general, when the hand was close to the cup, but without actually holding it. The few times when the Hand Wrist Module incorrectly detected the user as holding the cup, the system remained safe because the user was not looking at the cup.

Nonetheless, we cannot foresee all situations in which this can happen, e.g. the user might accidentally hit the cup, and so the hand/wrist would be in the circle area. This is why the pressure sensor scenarios proved to be safer. We used two thresholds one for each pressure sensor scenario. For the first one, the threshold was low, to permit the system to detect when the user was holding the cup without applying too much pressure. In this situation, the robot said *"Please hold the cup."* A few adjustments were made along the way to find the right threshold, but overall there were few failures. The difficulty was in tuning the system to release the cup at the right moment. Out of the 15 tests for this scenario during two tests the threshold was too low. The robot released the cup too soon which increased the likelihood of the cup being dropped (although that did not actually happen). During a further two tests the threshold was

---

[5] These thresholds represent the percentages by which the pressure sensor values should minimally vary from the point when the control value is read (when the robot hand reaches the release position) until the three second time interval for checking whether or not the user is holding the cup ends.

set too high, resulting in the cup being dropped because users took away their hand assuming the robot would not release the cup.

The higher threshold, used when the robot said *"Please grab the cup."* actually affected the liveness of the system. Some users stopped holding the cup because they thought that the robot would not release it. In addition, because each user grabbed the cup in a slightly different way, the threshold needed to constantly be adjusted. Although the users learned that they should apply more pressure in order for the robot to release the cup after 4-5 attempts (on average), the lower threshold seemed to be the right solution to reach a compromise between the safety and liveness properties of the system. For our "drink serving" robot, the tests revealed that the lower pressure sensor threshold keeps the system safe and live at the same time. However, if the robot were to handle some very dangerous chemicals, it would surely be more important for the system to be designed safer and maintaining liveness becomes a secondary concern.

It was concluded that the safety and liveness properties were appropriate. They provided a good balance between safety and usefulness for this setup and hence form a good basis for further research into methods to ensure safe HRI.

9 Conclusion and Follow-on Research Directions

Clearly, *trust* is a broad category and many elements should be taken into account when analysing the trustworthiness of a robot (e.g. aesthetics, ergonomy). This paper, nevertheless, is focusing on two important factors derived from the field of Design Verification, namely safety and liveness. The scenario discussed in this paper relates to a robot bartender and the trust elements which have been presented are specific to this context. The aim of this investigation was to develop an environment to study trust of HRI in terms of safety and liveness aspects. We designed a safe, yet useful system and presented a state diagram to better illustrate the core algorithm. We extracted the safety and liveness properties of the system and experimentally evaluated these. We concluded that this set of properties provides a good balance between safety and usefulness of our "drinks serving" robot and a good basis for further research.

Our next steps include the development of a formal description of the system including the associated safety and liveness properties with the aim to formally verify that indeed these properties hold. In our view, formal verification complements experimental validation in real-world scenarios. It helps developers to obtain greater confidence in the system's functional correctness.

Another direction for further research is to enable the robot to self-adjust its thresholds based on evaluating failed and successful user interactions. We are currently conducting two follow-on pilot projects that introduce machine learning into the robot bartender setup. The properties established in this paper serve as specification for both these projects. One project is focused on extending reinforcement learning techniques to work within the constraints determined by the safety and liveness properties, thus ensuring that any adaptations maintain the specified properties and hence stay within safe bounds. The other project investigates the use of a requirements-based testing methodology similar to what is used for the certification of avionics software [10] to qualify any proposed adaptations produced by the learning system. By deriving tests from the requirements specification we expect to obtain a set of tests that are "immune" to valid adaptations, i.e. adaptations that do not violate the specification. Adaptations that violate the existing specification may be used to prompt designers/users to investigate whether or not the specification should be modified to permit such adaptations in the future.

In the context of the above two projects we are also investigating the added value gained from introducing *soft* and *hard* constraints. For example, a soft constraint might be that the usual serving distance is about an arm's reach from the human's body, while a hard constraint might be that the robot must not touch or push the human. Hence, for someone with an injured shoulder bone, the serving position would be much closer than usual which would require the soft constraint to be overwritten by an adaptation while the hard constraint must not be violated.

Ultimately, we are working towards finding a compromise between adaptability and verifiability. Our longer term research objectives are to develop design and verification methods that ensure humans can trust adaptive robots by demonstrating that all adaptations remain useful and within safe bounds. This is a significant challenge because the powerful and well-developed methods of Design Verification used in application domains such as the aerospace industries require complete a priori knowledge of all circumstances and states that a system could enter during operation. This is hard enough for, say, a jet-engine control unit. The internal and environmental complexity of a robot operating in unstructured human-inhabited settings, however, renders these approaches completely useless in their current forms and requires significant innovation that can only be achieved through cross-disciplinary collaboration.

Acknowledgements: The authors acknowledge the help received from Minshuai Zhao, an MSc student collaborating on this project. We also thank Armando Tacchella for constructive feedback.

References

1. Basu, A., Gallien, M., Lesire, C., Nguyen, T.H., Bensalem, S., Ingrand, F., Sifakis, J.: Incremental component-based construction and verification of a robotic system. In: European Conference on Artificial Intelligence. pp. 631–635. IOS Press (2008)
2. Bensalem, S., Gallien, M., Ingrand, F., Kahloul, I., Nguyen, T.H.: Toward a More Dependable Software Architecture for Autonomous Robots. IEEE Robotics and Automation Magazine 16(1), 67–77 (2009)
3. Clarke, E.M., Grumberg, O., Peled, D.: Model checking. MIT Press (1999)
4. Fitzpatrick, P., Metta, G., Natale, L.: Towards long-lived robot genes. Robotics and Autonomous Systems 56(1), 29–45 (Jan 2008)
5. Lallée, S., Lemaignan, S., Lenz, A., Melhuish, C., Natale, L., Zant, T.V.D., Warneken, F., Dominey, P.F.: Towards a Platform-Independent Cooperative Human-Robot Interaction System: I. Perception. In: International Conference on Intelligent Robots and Systems (IROS 2010). pp. 4444–4451. IEEE (2010)
6. Lenz, A., Skacheck, S., Hamann, K., Steinwender, J., Pipe, A., Melhuish, C.: The BERT2 infrastructure: An integrated system for the study of human-robot interaction. In: International Conference on Humanoid Robots. IEEE (2010)
7. Magee, J., Kramer, J.: Concurrency: State Models & Java Programs. Wiley (2006)
8. Metta, G., Natale, L., Pathak, S., Pulina, L., Tacchella, A.: Safe and Effective Learning: A Case Study. In: ICRA 2010. pp. 4809–4814 (2010)
9. Pappas, G., Kress-Gazit, H. (eds.): ICRA Workshop on Formal Methods in Robotics and Automation (2009)
10. RTCA: DO178B Software Considerations in Airborne Systems and Equipment Certification (1992)