Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Grant Agreement Type: Collaborative Project
Grant Agreement Number: 611391

D6.3.2 Deliberative Subsystem

Due date: **1/4/2016**
Submission Date: **30/03/2016**

Start date of project: **01/04/2014** Duration: **54 months**

Organisation name of lead contractor for this deliverable: **Vrije Universiteit Brussel**

Responsible Person: **Bram Vanderborght** Revision: **1.3**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**PU**
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Executive Summary

Deliverable D6.3 defines the specification, design, implementation and validation of the Deliberative subsystem within the cognitive architecture in WP6. Specifically, this report presents the advances in task T6.3 for the first and second year of the DREAM project. During the first year the cognitive architecture and the Deliberative subsystem were designed. During the second year the focus was on the autonomous acquisition of action selection through machine learning.

Note: this is a living document and extends on the preliminary version of this deliverable which was submitted for review last year.

Principal Contributors

The main authors of this deliverable are as follows (in alphabetical order).

Paul Baxter, Plymouth University
Tony Belpaeme, Plymouth University
Hoang-Long Cao, Vrije Universiteit Brussel
Albert de Beir, Vrije Universiteit Brussel
Pablo Gómez, Vrije Universiteit Brussel
James Kennedy, Plymouth University
Emmanuel Senft, Plymouth University
Greet Van de Perre, Vrije Universiteit Brussel
Bram Vanderborght, Vrije Universiteit Brussel

Revision History

Version 1.0 (P.B. 17-02-2016)
Skeleton structure and appendices.

Version 1.1 (T.B. 19-02-2016)
Contents from previous preliminary deliverable report to build on.

Version 1.2 (T.B. 21-03-2016)
Added new papers published in Period 2 and added section 4.

Version 1.3 (J.K. 30-03-2016)
Further updates for Period 2.

1 Overview of WP6 Architecture

In DREAM we will move away from Wizard of Oz-controlled behaviour for the robot, which too often is the de facto mode of interaction in Robot Assisted Therapy [1]. Therefore work package WP6 aims to progress the theoretical and methodological understanding of how an embodied system can interact autonomously with young users in a learning task, specifically developed for atypically developing children. WP6 is concerned with the development of the robot behaviour subsystems to provide social robots with a behaviour underlying social interaction, which permits the robot to be used in Robot Enhanced Therapy (RET) in a supervised autonomous way. This involves both autonomous behaviour and behaviour created in supervised autonomy, whereby an operator requests certain interventions, which are then autonomously executed by the robot.

A general high level description of the robot control system is shown in Figure 1 (also see Annex 5.2). This basically describes how the autonomous controller is informed by three external sources: the child behaviour description, sensory information, current intervention script state, and input from a therapist (e.g. emergency stop, not shown in diagram). Combining these sources, the autonomous controller should trigger as an output the appropriate sequence of action primitives to be performed (as well as some feedback via the WoZ GUI), which then gets executed on the robot.

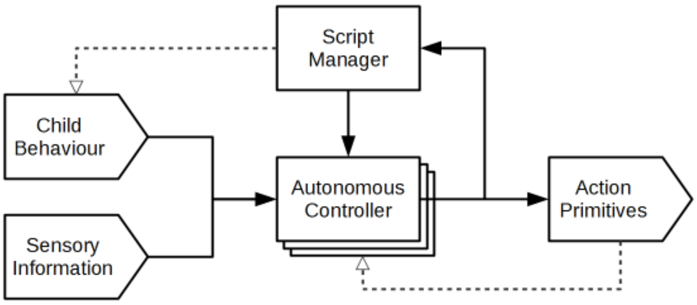

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied. The intervention script provides context for child behaviour interpretation.

The autonomous controller is composed of a number of subsystems, as described in the DoW: Reactive, Attention, Deliberative, Self-Monitor and Expression and Actuation. In the Reactive subsystem, sensory inputs are immediately acted upon with appropriate actuator outputs. The Attention subsystem determines the robot's focus of attention. In the Deliberative subsystem, the necessary interventions will be implemented in a general approach so it is not scenario-specific. The Self-Monitoring subsystem acts as an alarm system in two specifications. An internal one when the robot detects that it cannot act because of a technical limitation or an ethical issue. An external alarm is one where the therapist overrules the robot behaviour selection. Finally, the Expression and Actuation subsystem is responsible for generating believable human/animal-like smooth and natural motions and sounds that are platform independent. These subsystems interact, and must combine their suggested courses of actions to produce a coherent robot behaviour, in the context of constraints laid down by the therapist (for example, the script to be followed, types of behaviour not permissible for this particular child because of individual sensitivities, etc). As a result, we have formulated the following architecture describing how cognitive control informed by the therapy scripts is to be achieved (Figure

2), see Annex 5.2 for further details.

A detailed description of the cognitive architecture was provided in deliverable D6.1 at month 18. Within this report we describe the functionality of the Deliberative subsystem. The next version of this document will be ready for month 36.

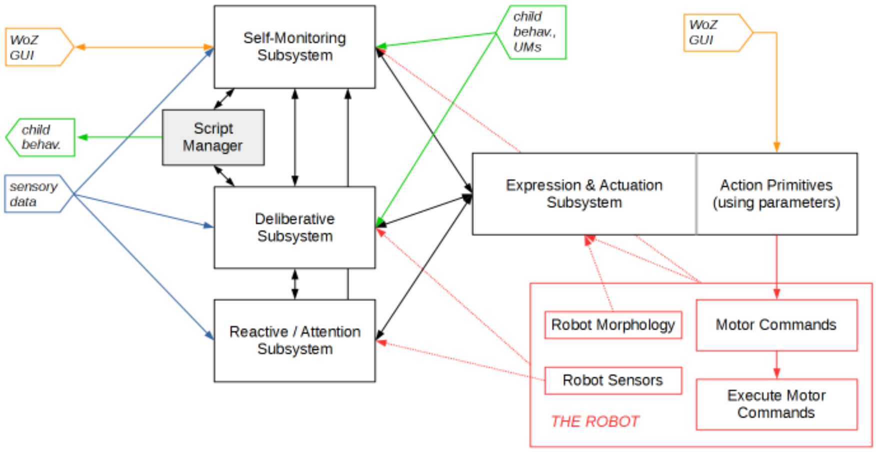

Figure 2: Description of the cognitive controller subsystems. The script manager is separate from, but tightly interacts with, the Deliberative subsystem to enable the robot control system to generate appropriate social/interaction behaviour even in the absence of an explicit interaction script. *UMs*: User Models.

2 The Deliberative Subsystem

The main goal for this subsystem is to make decisions on which behaviour has to be selected based on the requirements of the therapy; what the Attention subsystem is capturing from the surroundings; whether or not the child is motivated enough and how he or she is performing in each of the scenarios; and finally, the on-line feedback that the therapist could be providing through the WoZ GUI. Such behaviour will be sent to the Expression and Actuation subsystem.

2.1 Overview

A central aspect of the cognitive controller is its ability to follow intervention scripts as defined by the clinicians for both diagnosis and therapy. These scripts describe the high-level desired behaviour of the robot[1], and the expected reactions and behaviours of the child, in a defined order.

The decision was made to separate the script manager from the Deliberative subsystem itself (Figure 3). This decision was taken for a number of reasons. Firstly, it enables the cognitive control of the robot to be independent of the precise application domain - with the intention that the developments made would be more generally applicable within the field of social robotics, although the script-based behaviours remain a central part of the behaviour generation of the system. Secondly, it

[1] These predefined robot behaviours differ from the the low-level motor control of the robot, as these may be mixed with other aspects of behaviour not specified explicitly in the high-level intervention script; e.g. the addition of attention to unexpected events in the environment.

ensures that it would be possible to change the scripts in the future to alter their relative difficulty, by for example including further steps in the intervention, changing the type of intervention, or creating different activities, due to a modular design[2]. As a consequence of this, the Deliberative subsystem is now primarily focussed on action selection considerations, making use of a range of algorithms and methodologies as will be explored in the coming years. Thirdly, this division of the script manager from the Deliberative subsystem enables the system to generate coherent behaviour even if there is not a script active at a given moment. This could be useful for periods between the explicit intervention sessions for example, where the robot would then still be able to respond appropriately to environmental stimuli, if so desired by the therapists. These are consistent with the aims expressed within the WP6 Description of Work.

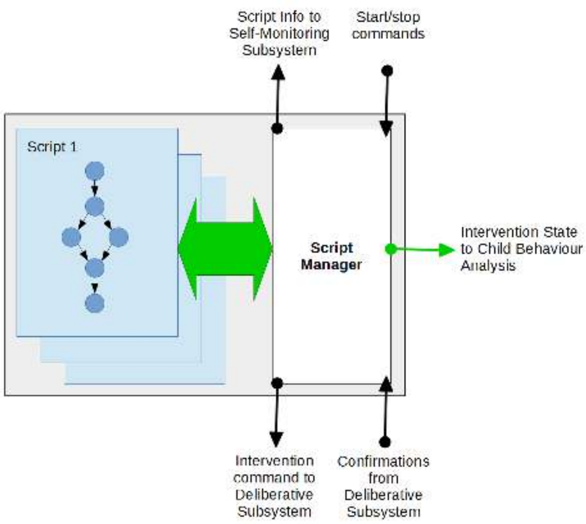

Figure 3: Overview of the script manager subsystem. The scripts are defined independently of the script manager, which is responsible for stepping through the script as appropriate and communicating with the other subsystems as required.

The script manager itself separates the logic necessary to manage progression through the script (by taking into account the available sensory feedback after actions for example) from the script itself. This makes it straightforward to add new scripts or modify existing scripts as required. This logic management could in the first instance be achieved using a Finite State Machine (FSM). Details of the preliminary script manager implementation may be found below in Section 2.2.

One possibility for the scripts is that each step in the script be defined as a 3-tuple of the form: *[existing_state, proposed_action, consequent_state]*. In this context, *existing_state* could be defined by default to be the *consequent_state* of the previous step. The *proposed_action* defines what action should be taken by the robot, and be one of the actions (or unique identifier thereof) defined in D1.2. The *consequent_state* defines what robot state should be expected (in terms of sensed state) if the *proposed_action* were successfully completed. This may be used by the script manager to determine if and when it is appropriate to move onto the next script step. These 3-tuples may initially be held in a plain text file to facilitate examination and modification by the clinical staff as required. This can be

[2]As noted above, these high-level scripts do not necessarily completely define the behaviour of the robot, and are distinct from any predefined robot motor control sequences that may be used, such as waving or nodding.

changed later to ease the process (for example by providing a drag-and-drop script construction GUI).

The Deliberative subsystem is the primary locus of autonomous action selection in the cognitive controller (Figure 2). This subsystem takes as input sensory data, child behaviour information, information on what step should be next executed from the therapy script, and higher-level direction from the Wizard/Self-Monitoring subsystem. It then proposes what action should be taken next by the robot (this proposal is sent to the Expression and Actuation subsystem). In a normal script execution context, the Deliberative subsystem is the primary driver of behaviour, which would typically propose the next script step. Details of the deliberative subsystem implementation may be found below in Section 2.2.

There are however a number of circumstances in which this is not the most appropriate action to perform. For example, if the child is detected to have very low engagement with the task (as determined from the WP5 component/s, and/or information from WP4 sensory system saying the child is looking away for example), then it would be appropriate to attempt to re-engage the child with the robot/task prior to executing the next stage in the therapy script. In this case, the Deliberative subsystem can choose to depart from the behaviour defined in the script, and instead propose a different behaviour.

2.2 Preliminary Deliberative Subsystem Component

Based on the functional description of the cognitive controller system of the DREAM architecture (see section above, and Annex 5.2), preliminary implementations of the deliberative subsystem and script manager components have been formulated.

These first versions of the components are defined in terms of the input and output ports, following the guidelines established in the software engineering standards (WP3). These preliminary versions are directly informed by the development of the WP6 control architecture in Y1, where each subsystem was defined in terms of the interactions with other subsystems, and their functions as outlined in the DREAM DoW. Please refer to Figure 1 to provide this context.

2.2.1 Script Manager

The functions of the script manager are described above: the main point is that the script manager is separated from the rest of the Deliberative subsystem, which is instead focussed on autonomous action

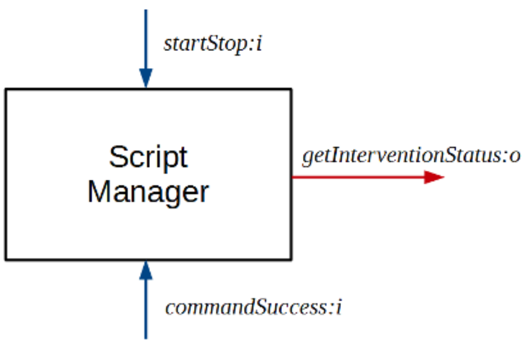

Figure 4: Script Manager component YARP ports: the prefix for the port names is listed in the main text. This component handles the script state and provides this information to other components.

selection. The ports for this component are described in Figure 4. As per the software engineering standards, each of the port names shown is prefixed by the system and subsystem names. For the script manager, this prefix is "*/cognitiveController/scriptManager/*".

The */getInterventionStatus:o* port provides the current state of the intervention, which can be accessed by any component that requires it (notably the WoZ interface, the deliberative subsystem, and the child behaviour recognition component). This port is of type *BufferedPort<VectorOf<Int> >*. The *startStop:i* port is used by the therapist GUI to indicate the start and end of the interaction. The *commandSuccess:i* port provides feedback that the last requested intervention command was successfully execeuted (or not). In this preliminary specification, both of these ports are of type *BufferedPort<Bottle>*.

2.2.2 Deliberative Subsystem

As with the script manager component, the Deliberative subsystem port names are prepended with the subsystem name, in this case "*/cognitiveController/deliberativeSubsystem/*". An overview of the ports for this component are shown in Figure 5.

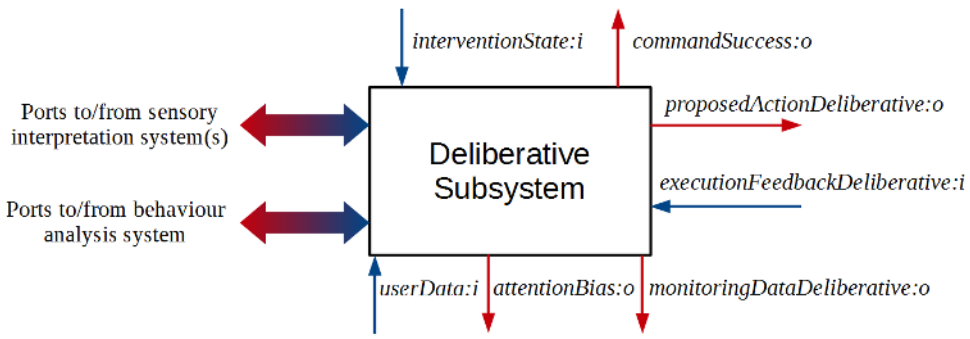

Figure 5: Deliberative subsystem component YARP ports: the prefix for the port names is listed in the main text. This component is responsible for the autonomous action selection in cases of deviation from the script, etc. Ports to/from the sensory systems and child behaviour analysis system are not shown for clarity.

For the sake of clarity, the ports to/from the child behaviour classification component (3 ports) and the sensory interpretation component (25 ports) are not shown in Figure 5: full details of these may be found in D3.1. The "*getInterventionStatus:i*" is of type *BufferedPort<VectorOf<Int> >*. In the preliminary version of the Deliberative subsystem, the remaining six ports are of type *BufferedPort<Bottle>*.

Further to the definition of this preliminary version of the YARP component to fit into the DREAM architecture according to the established software engineering framework (see D3.1), a range of work has been conducted on the theoretical basis of autonomous action selection mechanisms for social robots. In addition to the technical principles described below (Section 2.3), this development work has also included the principled examination of what sort of functionality should be undertaken by this component, and what the limitations of this should be with respect to the supervisory oversight provided by the ever-present therapist (as defined by the *supervised autonomy* objective of DREAM). Our preliminary work in this regards is summarised in Annexes 5 and 6.

2.3 Action Selection Mechanism

2.3.1 Context

The script provides a series of actions that the robot should execute and that are expected to be followed by a child-specific reaction. However, the actual reaction from the young user can be different to the one expected. To be able to follow the script, the robot needs to find a way to execute the appropriate behaviour in order to obtain the desired reaction from the child. To succeed in such a challenging task, a social Action Selection Mechanism (ASM) is required.

This social ASM has to fulfil multiple specifications: the first one is to allow the robot to detect a state where the execution of an action not planned in the script is required. This can be done by comparing the child's current state to the expected one. Then, once the need to act outside of the script is detected, the ASM has to select an action that would obtain the expected reaction from the child and thus allow continuation of the script.

Having to select an action in a Robot Assisted Therapy (RAT) scenario gives rise to several concerns. First of all, at every moment of the interaction, the action executed by the robot needs to be the correct one, i.e. the one desired to maximise the positive effect of the therapy. Furthermore, as we are working with children, the environment is highly unpredictable and there is probably not a unique general solution for every child and a solution for one child might not be appropriate later in the interaction: the robot needs to be able to adapt to different interaction partners and also to the same partner at different times in the interaction. Lastly, with RAT, we have access to experts with good knowledge of the environment (both the children and the task to be accomplished): the therapists. We can use the therapists to obtain some knowledge, but we can not rely totally on them as this would impose a high workload on them and this is not scalable.

Consequently, we end up with three principles that the ASM has to follow:

1. The action selected has to be therapeutically correct at every stage of the interaction.
2. The ASM has to be adaptive, i.e. be able to change the action policy over time.
3. The workload on the therapist should be as low as possible.

2.3.2 State of the Art

When having to design an ASM, the simplest possibility is to use a static predefined behaviour, such as a reactive system or a finite state machine, as has been used in some experiments of the Aurora project [2]. This has the advantage of not relying at all on the therapist, but as we are working in the real world, the sensors are represented in a high-dimensional and continuous space. This means that either the behaviour expressed would be simple, or that a more complex controller would probably not be able to be designed manually; there is no way to know in advance what the best action to perform in each state is. Similarly, as it is not possible to have a precise enough model of the child, pure planning is not suitable in our case.

The other main approach used in RAT is the Wizard of Oz (WoZ) paradigm [3, 4]. The robot is not autonomous, but is instead fully teleoperated by a therapist. This can fulfil principles 1 and 2 as the action is always the one that the therapist would desire and the human also provides adaptivity. However, as explained by Thill et al. in [1], there are many reasons which motivate us to move away from WoZ, such as the reliance on the humans violating principle 3 through the imposition of a heavy workload.

A way to fulfil principles 2 and 3 is to provide a robot with learning capabilities. If the robot can learn from successes and errors, it becomes adaptive and does not rely on a human to select its actions. A classical technique used to allow an agent to learn by itself when interacting with an environment is Reinforcement Learning (RL) [5]. With this approach, the robot explores its environment by interacting with it. Through receiving positive or negative rewards from its environment, it optimises an action policy after a learning phase, which can achieve high task performance. This technique could asymptotically also fulfil principle 1, but at the start of the interaction the robot relies on exploration to create its action policy. As such, random actions can be executed to obtain knowledge, and this has the potential to violate principle 1, with possible negative therapeutic outcomes. Furthermore, without external help, this method can take a long time to converge, and depends on rewards from the environment to learn, which might be hard to define explicitly for RAT.

Some researchers have worked on ways to improve RL, for example in [6], authors assume that the environment is not giving rewards, and that a human can give them. This could be suitable for our case as the therapist possesses the adequate knowledge to evaluate the robot actions. A similar approach has been followed by Thomaz et al. in [7] where they combined rewards from the environment, reward from a user and guidance from a user. This allows faster convergence toward an efficient policy and reduces the number of potentially incorrect actions. However in all of these methods, the evaluation is done a posteriori, so if an action is not correct, it will not be executed again, but it has still been executed once. In RAT, this is something we have to avoid as even one incorrect action could have negative consequences.

To cope with this exploration problem, some authors proposed Safe RL [8], in this case, an additional mechanism is combined with the RL policy to prevent dangerous actions from being executed. They present two main ways to make RL safer: using initial knowledge to prevent the execution of actions in specific cases, or to bootstrap the learning with safe demonstration, and to only explore around them. But even these methods can not guarantee that only correct actions will be executed as there is still a reliance on exploration and all of the potentially dangerous state-action pairs can not be defined in advance.

Another method proposed to achieve autonomy in RAT without having the robot to explore environment by itself is inspired by Learning from Demonstration and the WoZ paradigm. In [9], authors propose that the interaction is begun in a classical WoZ setup. The robot is only controlled by the therapist initially, then when enough data has been gathered, batch learning is applied and an action policy is derived. As only correct actions have been given to the robot, and if we assume that enough data points are obtained to cover the environment space, the action policy derived is correct. Only therapeutically valid actions will be executed, and this is achieved without further reliance on the therapist to control the robot. However, as no further learning is used, the ASM is not adaptive, so it can not cope with additional changes in the child behaviour. As such, even if principle 1 and 3 are fulfilled, principle 2 is still missing.

2.3.3 Proposed Solution

As shown previously, there is currently no solution in the literature fulfilling all of the desiderata for our Action Selection Mechanism. Methods either require important knowledge to be hard-coded inside the software, rely on exploration involving randomness, impose a heavy workload on the therapists, or are not adaptive once the learning is finished.

To be able to fulfil principles 2 and 3, the ASM has to include a machine learning component. This is the only way to provide the required adaptivity without relying on a human. However, the majority of the algorithms used for learning face a trade-off between exploitation and exploration: the agent

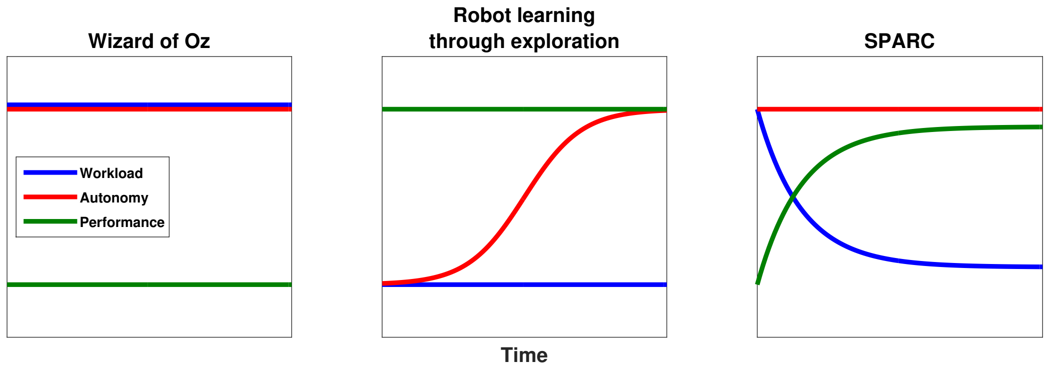

Figure 6: Comparison of SPARC with WoZ and Reinforcment Learning in terms of autonomy, supervisor workload, and performance. The aim is for workload to be low, whilst autonomy and performance are high.

first needs to explore its environment to be able to select the best action to perform later, and generally there is an element of randomness in the exploration. As encompassed by principle 1, in RAT there is no place for randomness. We need to find an ASM allowing the robot to become autonomous, to learn the best action to execute in a highly unpredictable, continuous and complex space without relying on random exploration.

As developers, we have limited knowledge of the best action to execute in an unexpected state; this knowledge is the expertise of the therapist. This expertise can be used to provide the initial knowledge for our learning mechanism. Furthermore, this knowledge can also prevent incorrect actions from being executed during the learning phase, or indeed in any part of the interaction.

This is the reason why we propose Supervised Progressively Autonomous Robot Competencies as a solution (SPARC; see Section 4). This technique relies on a system of suggestion/correction: the robot selects an action according to the ASM and suggests it to the supervisor (in this case, the therapist). In response, the therapist can either do nothing and thus the suggested action is executed, or select a different action for the robot to execute. This concept ensures that the right action is always executed: fulfilling principle 1. Simultaneous learning on the robot side allows the suggested action to be more appropriate with more interactions, reducing the workload on the therapist over time and fulfilling principles 2 and 3 (behavioural adaptivity and low therapist workload), whilst maintaining principle 1 (correct therapeutic action). This method is described in more detail in [10] and Section 4 presents the work done in evaluating this approach.

Figure 6 presents the concept of SPARC compared to WoZ or RL in terms of workload on the supervisor, autonomy and performance. With WoZ, at all times, there is no autonomy and a high workload on the supervisor providing a high performance. With RL, the human is not involved, so the autonomy is high and the workload is low. At the start, the robot is exploring the environment resulting in low task performance in the learning phase, and this performance rises until reaching an asymptotically high value once the robot knows how to act. SPARC imposes a high workload on the therapist at the start, when the robot is still learning. This provides faster learning, which allows the robot to be more autonomous with time and decreases the workload on the supervisor, whilst maintaining good task performance throughout the interaction.

2.4 Planned Work

In Period 1, the WP6 and Deliberative subsystem architectures were established from a theoretical basis, leading to the development of preliminary Deliberative subsystem and script manager components (see Sections 2.1 and 2.2). During this period, WP6 also supported manually controlled evaluation of the diagnosis/intervention scripts (defined in deliverable D1.1), further detail can be seen in Section 3 below. This provided guidance for further development within Period 2, which gave increased focus on how machine learning can be utilised to improve the autonomy of the robot in interactions (see Section 4 for details).

During Period 3, the Deliberative subsystem will be further developed in accordance with milestone MS4 specified within the WP6 Description of Work ("core functionality in robot behaviour"). Specifically, this will entail a completed version of the script manager component, as described in Sections 2.1 and 2.2. This will incorporate the intervention scripts as defined in deliverable D1.1, along with the logic to manage progression through the script. The cognitive architecture and robot behaviours developed in Periods 1 and 2 will facilitate the implementation of the autonomous version of the script manager component.

Section 2.3 described the approach to machine learning (SPARC) adopted within the context of the Deliberative subsystem that will strive to increase the autonomy of the robot behaviour. Successful evaluation of this approach in non-therapeutic environments in Period 2 (discussed in Annex 6.3 and Section 4) provides a solid platform for further exploration in Period 3. The SPARC model has already been tested with a Neural Network and Reinforcement Learning, with preliminary results suggesting promising performance when compared to WoZ or classical Interactive Reinforcement Learning (Section 4). Period 3 will see detailed analysis of these results and further exploration of an appropriate machine learning algorithm for the therapeutic application. The current SPARC implementation exists outside of the preliminary Deliberative subsystem component, so the placement of the SPARC approach into this component will be explored in Period 3. It is planned for the approach to be evaluated in settings more closely resembling the therapeutic environment of the DREAM project to provide additional validation prior to potential integration into the cognitive controller moving forwards.

3 Script Following

A primary objective of the first year of work was the evaluation of manually controlled ('wizarded' or tele-operated) versions of the diagnosis/intervention scripts defined in deliverable D1.1, as relevant to T2.1. This evaluation provides guidance for the further development of the autonomous interpretation and behaviour systems for the DREAM architecture.

In Y1, WP6 provided substantial support to provide the systems necessary for these evaluations. The robot behaviour capabilities to enable execution of each of the basic versions of the scripts has been implemented: imitation task, joint attention task, and the turn-taking task. These systems are currently deployed and in use at partner UBB.

Two methods were used to provide this functionality. For the imitation and joint attention tasks, behaviours were constructed in the Aldebaran-produced Choregraphe suite, such that a therapist could manually control the robot behaviours for each of the stages of the task. Details of this system can be found in Annex 5.4 of this deliverable. For the turn-taking task, since the Sandtray device is used, a standalone system using the software engineering standards defined in WP3 were used. Details of this system can be found in Annex 5.3 of this deliverable.

This work provided the basis for further developments within WP6 in Y2. The development of the behaviours for each of the intervention tasks can be reused in the autonomous versions of these tasks, along with further behaviours as required. Furthermore, the establishment of preliminary versions of the various components using the software engineering framework, and the development of the WP6 cognitive control architecture, will facilitate the implementation of the autonomous versions of these components, and their subsequent integration with the rest of the system in Y3 and beyond.

4 Increasing Robot Autonomy

In the second year of work there has been an increased focus on studying how machine learning can be used to gradually take over from the therapist, or in the broader sense of the word, the "Wizard". For this, a new method was developed, dubbed SPARC (Supervised Progressively Autonomous Robot Competencies). SPARC proposes actions to the supervisor and observes which actions the Wizard takes in which states; the states are comprised of internal states of the robot and external states in the social and physical environment, including the child. SPARC gradually builds up a state-action model, and as the interaction progresses, suggests more appropriate actions to the Wizard. The Wizard can relinquish control to SPARC by accepting its proposed actions.

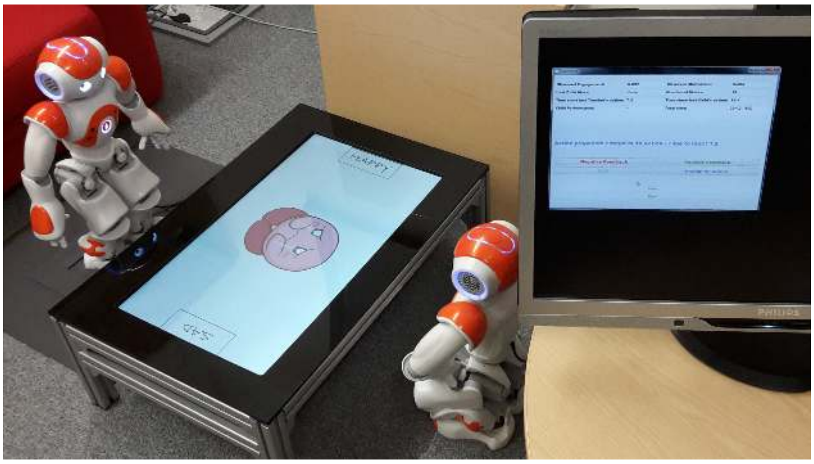

Figure 7: Setup used for the user study from the perspective of the human supervisor. The *child-robot* (left) stands across the touchscreen (centre-left) from the *wizarded-robot* (centre-right). The supervisor can oversee the actions of the *wizarded-robot* through the GUI and intervene if necessary (*right*).

In Period 2, the architecture has been further developed from the simulation model presented in Period 1. The SPARC model has been tested both with a Neural Network and Reinforcement Learning and has been developed to suit the context of Robot Assisted Therapy [10, 11]. A main focus has been on evaluating how human operators, or Wizards, perceive the gradually increasing autonomy of the robot and the impact on the task performance.

To this end we used a novel method, in which the child in the interaction is substituted by a robot running a "child" model (Figure 7). This allows experimenting without putting undue pressure and stress on young participants, and provides a setup with high repeatability, which is required for rigorously testing the SPARC architecture. A number of hypotheses were evaluated in a study [10]. Overall, the study showed that controlling a learning robot enables supervisors to achieve similar task performance as with a non-learning robot, but with both fewer interventions and a reduced perception of workload. These results demonstrate the utility of the SPARC concept and its potential effectiveness to reduce the cognitive and workload on human operators.

SPARC has recently been implemented in a restricted (non-therapeutic) environment and compared to previous work done in Interactive Reinforcement Learning (IRL): the environmental reward is combined with reward given by the user after the action execution. Preliminary analysis of the results indicates that SPARC is compatible with Reinforcement Learning, and it leads to faster and better results than classical IRL with a lower workload on the supervisor. These results will be analysed in greater detail and will form the basis for further exploration of this approach in year 3.

References

[1] Serge Thill, Cristina A Pop, Tony Belpaeme, Tom Ziemke, and Bram Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, 2012.

[2] Kerstin Dautenhahn. Robots as social actors: Aurora and the case of autism. In *Proc. CT99, The Third International Cognitive Technology Conference, August, San Francisco*, volume 359, page 374, 1999.

[3] Jelle Saldien, Kristof Goris, Bram Vanderborght, Johan Vanderfaeillie, and Dirk Lefeber. Expressing emotions with the social robot probo. *International Journal of Social Robotics*, 2(4):377–389, 2010.

[4] Ben Robins, Kerstin Dautenhahn, and Paul Dickerson. From isolation to communication: a case study evaluation of robot assisted play for children with autism with a minimally expressive humanoid robot. In *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*, pages 205–211. IEEE, 2009.

[5] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT Press Cambridge, 1998.

[6] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.

[7] Andrea L Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716–737, 2008.

[8] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

[9] W Bradley Knox, Samuel Spaulding, and Cynthia Breazeal. Learning social interaction from the wizard: A proposal. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[10] Emmanuel Senft, Paul Baxter, James Kennedy, and Tony Belpaeme. Sparc: Supervised progressively autonomous robot competencies. In *Social Robotics*, pages 603–612. Springer, 2015.

[11] Emmanuel Senft, Paul Baxter, and Tony Belpaeme. Human-guided learning of social action selection for robot-assisted therapy. In *4th Workshop on Machine Learning for Interactive Systems*, 2015.

5 Period 1 Annexes

5.1 Senft, E. et al. (2015), When is it better to give up? Towards autonomous action selection for robot assisted ASD therapy

Bibliography - Senft, E., Baxter, P., Kennedy, J., Belpaeme, T (2015), "When is it better to give up? Towards autonomous action selection for robot assisted ASD therapy", HRI'15 Extended Abstracts, doi: 10.1145/2701973.2702715

Abstract - Robot Assisted Therapy (RAT) for children with ASD has found promising applications. In this paper, we outline an autonomous action selection mechanism to extend current RAT approaches. This will include the ability to revert control of the therapeutic intervention to the supervising therapist. We suggest that in order to maintain the goals of therapy, sometimes it is better if the robot gives up.

Relation to WP - This work directly contributes to Task T6.3.

5.2 Baxter, P. et al. (2015), Technical Report: Organisation of Cognitive Control and Robot Behaviour

Abstract - The purpose of this technical report is to summarise the motivations and constraints underlying the cognitive control structures, and to outline an organisation of these subsystems. This is a proposal only; this document is intended to be a working one, to be updated as required during development. This version of the report is based primarily on the discussions that took place in Brussels (23/01/15).

Relation to WP - This work directly contributes to Task T6.3.

5.3 Baxter, P. et al. (2015), Technical Report: Sandtray Wizard-of-Oz System for Turn-taking Intervention

Abstract - In this technical report we describe the software organisation of the Sandtray system created for the turn-taking diagnosis/intervention interactions. This system is based on the organisation defined by the WP3 software engineering standards, although at the moment does not fit into the rest of the DREAM system: this was to facilitate ease of setup and launch for the end-user (i.e. minimal installation, and no compilation required). The WoZ system provides a GUI from which the therapist can control the robot behaviour in the turn-taking task, and logs of the interaction are automatically stored for retrospective analysis.

Relation to WP - This work provides the basis of work in Task T6.3, and is relevant to T2.1.

5.4 Esteban, P.G. et al. (2015), Technical Report: Manual for the use of Choregraphe boxes in Wizard of Oz experiments

Abstract - In this technical report we describe a manual to help UBB team in the development of the Wizard of Oz experiments within Work Package 2. Both PLYM and VUB have collaborated to develop the corresponding modules in Choregraphe. This manual aims at being a reference point to ease the habituation of the therapists to the software.

Relation to WP - This work provides the basis of work in Task T6.3, and is relevant to T2.1.

6 Period 2 Annexes

6.1 Baxter, P. et al. (2015), Touchscreen-Mediated Child-Robot Interactions Applied to ASD Therapy

Bibliography - Baxter, P., Matu, S., Senft, E., Costescu, C., Kennedy, J., David, D., and Belpaeme, T. (2015) Touchscreen-Mediated Child-Robot Interactions Applied to ASD Therapy. New Friends symposium, Almere, The Netherlands.

Abstract - Robots are finding increasing application in the domain of ASD therapy as they provide a number of advantageous properties such as replicability and controllable expressivity. In this abstract we introduce a role for touchscreens that act as mediating devices in therapeutic robot-child interactions. Informed by extensive work with neurotypical children in educational contexts, an initial study using a touchscreen mediator in support of robot assisted ASD therapy was conducted to examine the feasibility of this approach, in so doing demonstrating how this application provides a number of technical and potentially therapeutic advantages.

Relation to WP - This paper summarises our use of touchscreen devices as mediating devices in child-robot interaction, and its specific use in diagnosing ASD.

6.2 Senft, E. et al. (2015) Human-Guided Learning of Social Action Selection for Robot-Assisted Therapy

Bibliography - Senft, E., Baxter, P., and Belpaeme, T. (2015). Human-guided learning of social action selection for robot-assisted therapy. In 4th Workshop on Machine Learning for Interactive Systems.

Abstract - This paper presents a method for progressively increasing autonomous action selection capabilities in sensitive environments, where random exploration-based learning is not desirable, using guidance provided by a human supervisor. We describe the global framework and a simulation case study based on a scenario in Robot Assisted Therapy for children with Autism Spectrum Disorder. This simulation illustrates the functional features of our proposed approach, and demonstrates how a system following these principles adapts to different interaction contexts while maintaining an appropriate behaviour for the system at all times.

Relation to WP - This paper sketches the early ideas on progressively learning autonomous behaviour from a human Wizard.

6.3 Senft, E. et al. (2015) SPARC: Supervised Progressively Autonomous Robot Competencies

Bibliography - Senft, E., Baxter, P., Kennedy, J., and Belpaeme, T. (2015). SPARC: Supervised Progressively Autonomous Robot Competencies. In Social Robotics (pp. 603-612). Springer International Publishing.

Abstract - The Wizard-of-Oz robot control methodology is widely used and typically places a high burden of effort and attention on the human supervisor to ensure appropriate robot behaviour, which may distract from other aspects of the task engaged in. We propose that this load can be reduced by enabling the robot to learn online from the guidance of the supervisor to become progressively more autonomous: Supervised Progressively Autonomous Robot Competencies (SPARC). Applying this concept to the domain of Robot Assisted Therapy (RAT) for children with Autistic Spectrum Disorder, a novel methodology is employed to assess the effect of a learning robot on the workload of the human supervisor. A user study shows that controlling a learning robot enables supervisors to achieve similar task performance as with a non-learning robot, but with both fewer interventions and a reduced perception of workload. These results demonstrate the utility of the SPARC concept and its potential effectiveness to reduce load on human WoZ supervisors.

Relation to WP - This paper describes the SPARC architecture, which can learn which actions to take in which states by observing a Wizard. The paper also presents a first user study which validates the concept.

6.4 Senft, E. et al. (2016) Providing a Robot with Learning Abilities Improves its Perception by Users

Bibliography - Senft, E., Baxter, P., Kennedy, J., Lemaignan, S. and Belpaeme, T. (2016) Providing a Robot with Learning Abilities Improves its Perception by Users. In Proceedings of the 11th Annual ACM/IEEE International Conference on Human-Robot Interaction, Christchurch, New Zealand.

Abstract - Subjective appreciation and performance evaluation of a robot by users are two important dimensions for Human- Robot Interaction, especially as increasing numbers of people become involved with robots. As roboticists we have to carefully design robots to make the interaction as smooth and enjoyable as possible for the users, while maintaining good performance in the task assigned to the robot. In this paper, we examine the impact of providing a robot with learning capabilities on how users report the quality of the interaction in relation to objective performance. We show that humans tend to prefer interacting with a learning robot and will rate its capabilities higher even if the actual performance in the task was lower. We suggest that adding learning to a robot could reduce the apparent load felt by a user for a new task and improve the users evaluation of the system, thus facilitating the integration of such robots into existing work flows.

Relation to WP - This study looks into how an operator (a Wizard) subjectively experiences a robot which gradually learns and takes over the operator's task.

6.5 Baxter, P. et al. (2016) Cognitive Architectures for Social Human-Robot Interaction

Bibliography - Baxter, P., Lemaignan, S. and Trafton, G. (2016) Cognitive Architectures for Social Human-Robot Interaction. In Workshop on Cognitive Architectures in Human-Robot Interaction, at the 11th Annual ACM/IEEE International Conference on Human-Robot Interaction, Christchurch, New Zealand.

Abstract - Social HRI requires robots able to use appropriate, adaptive and contingent behaviours to form and maintain engaging social interactions with people. Cognitive Architectures emphasise a generality of mechanism and application, making them an ideal basis for such technical developments. Following the successful first workshop on Cognitive Architectures for HRI at the 2014 HRI conference, this second edition of the workshop focusses specifically on applications to social interaction. The full-day workshop is centred on participant contributions, and structured around a set of questions to provide a common basis of comparison between different assumptions, approaches, mechanisms, and architectures. These contributions will be used to support extensive and structured discussions, with the aim of facilitating the development and application of cognitive architectures to social HRI systems. By attending, we envisage that participants will gain insight into how the consideration of cognitive architectures complements the development of autonomous social robots

Relation to WP - A position paper framing the need and state-of-the-art in cognitive architectures for social HRI, relevant to the deliberative subsystem in WP6.

6.6 Baxter, P. (2016) Memory-Centred Cognitive Architectures for Robots Interacting Socially with Humans

Bibliography - Baxter, P. (2016) Memory-Centred Cognitive Architectures for Robots Interacting Socially with Humans. In Workshop on Cognitive Architectures in Human-Robot Interaction, at the 11th Annual ACM/IEEE International Conference on Human-Robot Interaction, Christchurch, New Zealand.

Abstract - The Memory-Centred Cognition perspective places an active association substrate at the heart of cognition, rather than as a passive adjunct. Consequently, it places prediction and priming on the basis of prior experience to be inherent and fundamental aspects of processing. Social interaction is taken here to minimally require contingent and co-adaptive behaviours from the interacting parties. In this contribution, I seek to show how the memory-centred cognition approach to cognitive architectures can provide an means of addressing these functions. A number of example implementations are briefly reviewed, particularly focusing on multi-modal alignment as a function of experience-based priming. While there is further refinement required to the theory, and implementations based thereon, this approach provides an interesting alternative perspective on the foundations of cognitive architectures to support robots engage in social interactions with humans.

Relation to WP - A paper providing theoretical insights in how associative memories can serve as the backbone of a cognitive architecture. This approach is at present not implemented in DREAM, but is being explored in the context of the SPARC architecture.

Period 1

When is it better to give up?

Towards Autonomous Action Selection for Robot Assisted ASD Therapy

Emmanuel Senft, Paul Baxter, James Kennedy, Tony Belpaeme
Centre for Robotics and Neural Systems, Cognition Institute
Plymouth University, U.K.
{emmanuel.senft, paul.baxter, james.kennedy, tony.belpaeme}@plymouth.ac.uk

ABSTRACT

Robot Assisted Therapy (RAT) for children with ASD has found promising applications. In this paper, we outline an autonomous action selection mechanism to extend current RAT approaches. This will include the ability to revert control of the therapeutic intervention to the supervising therapist. We suggest that in order to maintain the goals of therapy, sometimes it is better if the robot gives up.

Categories and Subject Descriptors: H.1.2 [Models and Principles]: User/Machine System

Keywords: Action selection, ASD, Cognitive Robotics, RAT, Social Robotics.

1. INTRODUCTION

Recent studies estimate that around 1.1% of the population in the UK and also in other European countries have Autism Spectrum Disorders (ASD). These people typically lack social skills normally expected in human interactions. Consequently, therapies have been designed to help children with ASD to improve their social abilities; these therapies can be enhanced by using robots [5].

However, due to the complexity of social interactions involving children, the majority of existing studies use the Wizard of Oz (WoZ) technique, where the robot is not autonomous but controlled by a human. Despite the clear advantages of this method, there are a number of reasons for researchers to move away from it, such as reducing the personnel required to use the robots, or improve the consistency of therapy [3, 6].

The present work is conducted within the DREAM project: a European project which aims to develop new Robot-Enhanced Therapy. We seek to develop the therapy robot's control system to enable supervised autonomous operation. A clinician will set the therapeutic goal for the session, from which the robot should be able to decide by itself which actions to execute, under explicit supervision. Rather than maintaining autonomy, we argue that allowing the robot to revert control to the therapist when appropriate would improve both the interaction and the therapeutic outcome.

HRI'15 Extended Abstracts, March 2–5, 2015, Portland, OR, USA.
ACM 978-1-4503-3318-4/15/03.
http://dx.doi.org/10.1145/2701973.2702715.

Figure 1: The Aldebaran Nao was selected as the common robot platform, to facilitate consistency and reproducibility.

2. BACKGROUND

Different approaches of Robot Assisted Therapy (RAT) have been explored by researchers in the last two decades. In previous studies robot control was typically achieved in one of two ways: either fully tele-operated using the WoZ method, e.g. [4, 7] or fully autonomous, e.g. [1, 2]. For WoZ control, a hidden, manual manipulation of the robot allows the therapist to obtain exactly the desired behaviour and to adapt to unpredicted events. On the other hand, an autonomous robot requires lower load on a human operator and allows greater repeatability of behaviour, but requires to design a complex controller. As such, only reactive control schemes are used in prior work.

Several attempts have already been made to combine the flexibility offered by the WoZ method and the autonomy and the consistency provided by autonomous operation. However, working with children with ASD presents additional challenges: the infrastructure required to perform the experiments is more extensive, it is hard to gather a population large enough to obtain statistically valid results, therapies take place over long periods of time and, as ASD is a spectrum, the children's behaviour can be more difficult to predict than neurotypical children.

To be able to use a robot as a therapeutic tool, we use a set of interaction scripts that determine the interaction between the child and the robot. These scripts are defined and selected by a therapist according to the goals of the current session and describe a clear, serial interaction where both the robot's and the expected child's actions are specified.

However, as we are working with children with ASD, it is unlikely that the script will be completely adhered to. The robot needs to be able to react to unpredicted actions to either return to the script or find alternate means of continuing the interaction. An autonomous action selection mechanism must therefore be able to cope with unplanned events whilst maintaining the therapeutic goals for the current session. A general description of the context in which the action selection mechanism should operate is presented in Figure 2.

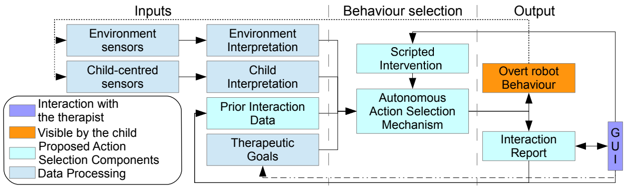

Figure 2: Context for the action selection mechanism.

Despite this, in some cases, the best action to select may be to stop the interaction and request help from the therapist. Contrary to the current approach, where the clinician must detect a problem and stop the interaction themselves, we aim to allow the robot to autonomously decide to refer to a human when this is appropriate.

3. APPROACH AND METHODOLOGY

The requirement for help for the robot may arise for a number of reasons. Firstly, there could be physical danger for the robot or the child. Some movements required by the script could harm the child if he is too close, requiring intervention if automated attempts to prevent collision fail.

Secondly, children could also react strongly to some specific actions, and force the therapist to stop the interaction. The robot could identify these actions and require help if one of them is requested for execution, thereby adjusting its level of autonomy. As the robot will be use in therapies, every action has to be carefully therapy-oriented and some actions could be defined that require approval before each execution.

Finally, the interaction could also fail (e.g. child no longer engages with the robot), where the robot does not have the competencies to pursue the interaction. In this case, therapist intervention would be required. If the therapist and the robot behave consistently in this context, the interaction may be more effective in terms of therapy.

3.1 Action Selection Mechanism

Based on previous studies about action selection in robotics, we have identified two broad approaches which could enable control to revert to the therapist. The first one is using a rule-based mechanism: as soon as a specific state is reached the therapist is consulted. An example is using the child's engagement in the interaction as a homeostatic variable: as soon as the implication goes outside a certain region, the interaction is stopped.

Another possibility is to use a predictive mechanism. Based on its previous interactions with this specific child or also with other children, the robot could have a model of what reaction is expected from its actions, and use it to predict the consequences of stopping the interaction versus continuing with its behaviour.

3.2 Evaluation methodology

To test our approach, we will use our algorithm in real session both with neurotypical children and ones with ASD in three scenarios: turn taking, imitation, and joint attention. If the robot detects a case where it has to revert the control to the therapist, it will broadcast a message describing what action should have been executed and why it stopped. The therapist will have the opportunity to execute the action, select another one, intervene in the interaction, or stop the session. We will use the therapist's action after control reverting and the number of times a therapist has to interrupt the interaction without a robot's prompt to evaluate the efficiency of the action selection mechanism.

4. DISCUSSION

Even when triggered by the robot, an unplanned human intervention in the interaction may have consequences on the child, the robot, and the therapist. For example, allowing autonomous failure detection, the robot can learn about it, and find itself a way to avoid the same state in the future.

Concerning the child, even if the session stops before an important problem, the emotional impact of interrupting the current interaction need to be taken into account. As children are sensitive, it is important to think carefully about the way to communicate the robot's failure to the child. Should the information about the interruption come from the robot? Should the therapist explain to the child what happened to the robot? We have no general solution yet, and the solutions may depend on individual characteristics. These questions have to be addressed based on data from empirical studies and collaboration with therapists.

5. CONCLUSIONS

In this paper we propose an approach to RAT for children with ASD: allowing a robot to voluntarily interrupt the interaction with a child and request help from a therapist. We outlined our motivations for this behaviour and presented possible consequences and questions to be resolved. The proposal is that autonomous action selection supports RAT because it reduces the workload on therapists, and improves its consistency.

6. ACKNOWLEDGMENTS

This work is supported by the EU FP7 DREAM project (grant 611391).

7. REFERENCES

[1] K. Dautenhahn. Robots as social actors: Aurora and the case of autism. *Proc. CT99*, (3), 1999.

[2] D. Feil-Seifer and M. Mataric. B3IA: A control architecture for autonomous robot-assisted behavior intervention for children with Autism Spectrum Disorders. *RO-MAN 2008.*, 2008.

[3] L. Riek. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, 1(1):119–136, Aug. 2012.

[4] B. Robins, K. Dautenhahn, R. T. Boekhorst, and A. Billard. Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills? *Universal Access in the Information Society*, 4(2):105–120, July 2005.

[5] B. Robins, K. Dautenhahn, and P. Dickerson. From Isolation to Communication: A Case Study Evaluation of Robot Assisted Play for Children with Autism with a Minimally Expressive Humanoid Robot. *Conferences on Advances in Computer-Human Interactions*, pages 205–211, Feb. 2009.

[6] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, and B. Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, Apr. 2013.

[7] B. Vanderborght, R. Simut, and J. Saldien. Using the social robot probo as a social story telling agent for children with ASD. *Interaction Studies*, 110(Tager 2001), 2012.

Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

TECHNICAL REPORT
Organisation of Cognitive Control and Robot Behaviour

Date: **02/02/2015**

Technical report lead partner: **Plymouth University**

Primary Author: **P. Baxter**

Revision: **2.2**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Service)	**PP**
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Summary

The purpose of this technical report is to summarise the motivations and constraints underlying the cognitive control structures, and to outline an organisation of these sub-systems. This is a proposal only; this document is intended to be a working one, to be updated as required during development. This version of the report is based primarily on the discussions that took place in Brussels (23/01/15).

Principal Contributors

The main authors of this document are as follows (in alphabetical order).

Paul Baxter, Plymouth University
Tony Belpaeme, Plymouth University
Hoang-Long Cao, VUB
Albert De Beir, VUB
Pablo Gomez, VUB
Emmanuel Senft, Plymouth University
Greet Van de Perre, VUB
Bram Vanderborght, VUB

Revision History

Version 1.0 (P.B. 21-01-2015)
Initial outline of ideas for the DREAM supervised autonomous robot control.

Version 2.0 (P.B. 26-01-2015)
Updated after discussions during Brussels meeting 23rd Jan 2015.

Version 2.1 (P.B. 02-02-2015)
Clarifications and updates following further discussion.

Version 2.2 (P.G. 27-02-2015)
Some modifications regarding priority system and Expression and Actuation subsystem functionality.

1 Aims and Constraints

An attempt is made in this section to formulate what the ideal (semi) autonomous system should conform to in terms of both clinical outcomes (i.e. the requests from the psychologists to improve the outcomes of individual children through robot-assisted therapy) and potential research (where this does not conflict with the clinical objectives).

The primary goal of the work in WP6 is to provide robot behaviour to facilitate the Robot-Assisted Therapy, see [1]. The main visible outcome of this should be the ability of the robot to execute the evaluation and therapeutic scripts as defined by the therapists. Whilst this must be achieved to fulfil the aims of the project, there are a number of areas in which there would be a role for behavioural adaptation, learning, and autonomous decision making. These should not however conflict in any way with the therapeutic goals for any given interaction session - indeed, it is necessary to vary the degree of shared control between the autonomous behaviour and the wizard supervisory control if this is more appropriate for a given child and/or circumstance.

Primary among these is the high probability that the interaction (due to the behaviour of the child for example) will deviate from the script. This must be handled in a manner consistent with the therapy, to not upset the child, and possibly (depending on the context) trying to re-engage the child with the script. A range of strategies will be required to deal with these situations, depending on the individual child (his/her characteristics) and the actual context for the departure from the script. This behaviour is likely to require flexible action selection, and will therefore require substantial research effort.

A second reason is that the robot is to demonstrate social behaviour in a supervised autonomous manner (with the requirement that the supervisor may over-rule this autonomous social behaviour if required). Social behaviour requires behaviour that is adaptive to the interaction partner in a range of interaction modalities (e.g. movement and speech). The autonomous behaviour of the robot must therefore be responsive to this, in a manner that is not, and indeed can not, be predetermined in the script.

Thirdly, given the range of intervention scripts that have been defined, there is also a possible need to modify the relative difficulty of the task (and/or interaction) given the specific characteristics and performance of the interacting child. This would, for example, involve varying the number and type of social behavioural cues used, the complexity of the required motor behaviours to complete the task, and/or the number of steps in the task.

The interfaces of the cognitive controller (WP6) with the rest of the DREAM integrated system (WP's 4 and 5) have already been defined. The intention in providing this overview document is to show how the subsystems of WP6 fit together to determine the behaviour of the robot in therapy interactions: the context in which each subsystem must operate is thereby defined. Initially, the skeleton of this system will be implemented in the most straightforward manner possible (with simplified code implementations of full component functionality for example) to check that the system fulfils all the requirements. This skeleton can then be filled in with more appropriate functionality over the course of the project.

2 Overall Organisation

A general high level description of the robot control system is shown is figure 1. This basically describes how the autonomous controller is informed by three external sources: the child behaviour description, sensory information, current intervention script state, and input from a therapist (e.g.

emergency stop, not shown in diagram). Combining these sources, the autonomous controller should trigger as an output the appropriate sequence of action primitives to be performed (as well as some feedback via the WoZ GUI), which then gets executed on the robot.

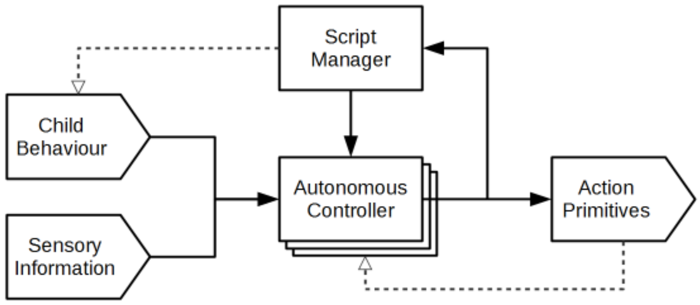

Figure 1: High level description of the robot control system. Child behaviour interpretation (WP5) and sensory information (WP4) provide the context for the autonomous action selection (as well as feedback from motor command execution), in combination with the particular intervention script being applied. The intervention script provides context for child behaviour interpretation.

The autonomous controller is composed of a number of sub-systems, as described in the DoW: Reactive, Attention, Deliberative, Self-Monitor and Expression and Actuation. These sub-systems interact, and must combine their suggested courses of actions to produce a coherent robot behaviour, in the context of constraints laid down by the therapist (for example, the script to be followed, types of behaviour not permissible for this particular child because of individual sensitivities, etc). An additional challenge is to ensure that the resulting system is independent of specific robot platform. As a result, we have formulated the following architecture describing how cognitive control informed by the therapy scripts is to be achieved (figure 2), an outcome of the WP6 meeting in Brussels (23/01/14). The following sections provide some further outline details of the main subsystems.

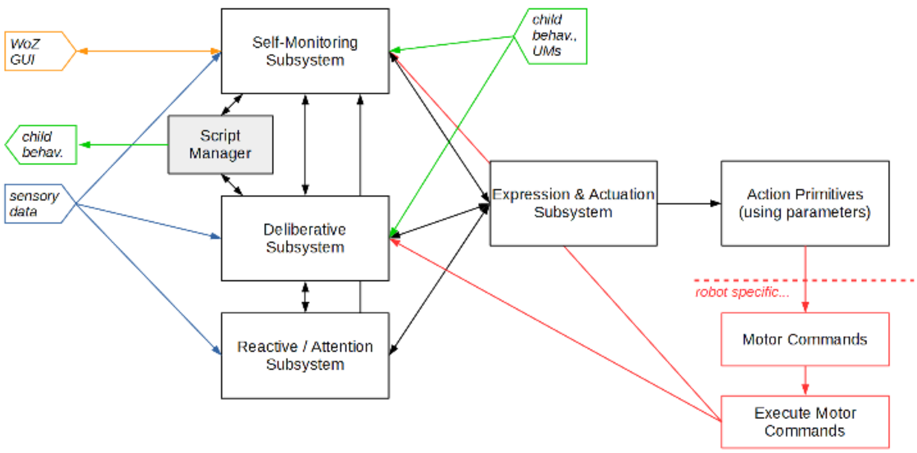

Figure 2: Description of the cognitive controller subsystems. The script manager is separate from, but tightly interacts with, the deliberative subsystem to enable the robot control system to generate appropriate social/interaction behaviour even in the absence of an explicit interaction script. UMs: User Models.

3 Reactive/Attention Subsystem

In the DoW, these are separated into two distinct subsystems. The reactive subsystem provides the general life-like behaviour of the robot (small motions, eye blinking, balancing, recovering from falls, 'pain' reactions, etc) in an as appropriate manner as possible (possibly requiring pilot studies to verify this). However, it should be possible to turn off these behaviours should the therapist deem it necessary for a particular child. This functionality is not envisaged to involve learning or adaptation. The attention subsystem is a "*combination of perceptual attention ... and attention emulation*". Making eventual use of saliency maps and habituation filters, this functionality will be guided by the deliberative subsystem.

We instead propose that these two subsystems be combined into a single component, due to the significantly overlapping technical systems required to fulfil the functions required. Both subsystems require access to features of the environment and interacting person(s) to respond appropriately (e.g. looking at a face or diverting attention to a loud noise somewhere in the environment). Managing this in a single component therefore seems a sensible choice so that functionality is not replicated. As planned in the DoW, it will be possible for the supervising therapist to switch off these functionalities if required for interaction with a particular child.

4 Deliberative Subsystem

A central aspect of the cognitive controller is the ability to follow intervention scripts as defined by the clinicians for both diagnosis and therapy. These scripts describe the high-level desired behaviour of the robot[1], and the expected reactions and behaviours of the child, in a defined order.

The decision was made to separate the script manager from the deliberative subsystem itself (fig 3). This decision was taken for a number of reasons. Firstly, it enables the cognitive control of the robot to be independent of the precise application domain - with the intention that the developments made would be more generally applicable within the field of social robotics, although the script-based behaviours remain a central part of the behaviour generation of the system. Secondly, it ensures that it would be possible to change the scripts in the future to alter their relative difficulty, by for example including further steps in the intervention, changing the type of intervention, or creating different activities, due to a modular design[2]. As a consequence of this, the deliberative subsystem is now primarily focussed on action selection considerations, making use of a range of algorithms and methodologies as will be explored in the coming years. Thirdly, this division of the script manager from the deliberative subsystem enables the system to generate coherent behaviour even if there is not a script active at a given moment. This could be useful for periods between the explicit intervention sessions for example, where the robot would then still be able to respond appropriately to environmental stimuli, if so desired by the therapists. These are consistent with the aims expressed within the WP6 DoW.

The script manager itself separates the logic necessary to manage progression through the script (by taking into account the available sensory feedback after actions for example) from the script itself. This makes it straightforward to add new scripts or modify existing scripts as required. This logic management could in the first instance be achieved using a Finite State Machine (FSM).

[1]These predefined robot behaviours differ from the the low-level motor control of the robot, as these may be mixed with other aspects of behaviour not specified explicitly in the high-level intervention script; e.g. the addition of attention to unexpected events in the environment.

[2]As noted above, these high-level scripts do not necessarily completely define the behaviour of the robot, and are distinct from any predefined robot motor control sequences that may be used, such as waving or nodding.

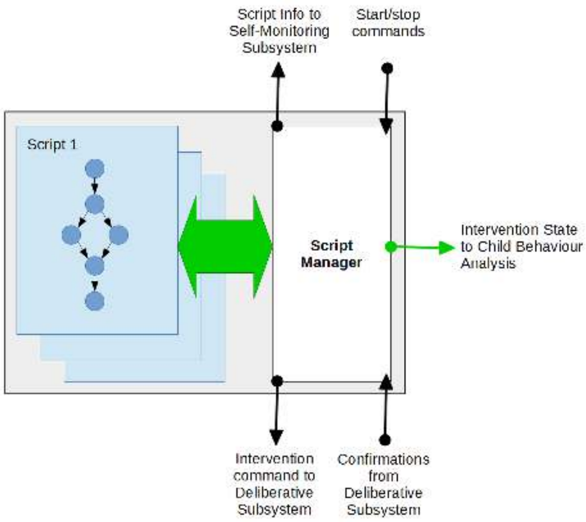

Figure 3: Overview of the script manager subsystem. The scripts are defined independently of the script manager, which is responsible for stepping through the script as appropriate and communicating with the other subsystems as required.

One possibility for the scripts is that each step in the script be defined as a 3-tuple of the form: *[existing_state, proposed_action, consequent_state]*. In this context, *existing_state* could be defined by default to be the *consequent_state* of the previous step. The *proposed_action* defines what action should be taken by the robot, and be one of the actions (or unique identifier thereof) defined in D1.2. The *consequent_state* defines what robot state should be expected (in terms of sensed state) if the *proposed_action* were successfully completed. This may be used by the script manager to determine if and when it is appropriate to move onto the next script step. These 3-tuples may initially be held in a plain text file to facilitate examination and modification by the clinical staff as required. This can be changed later to ease the process (for example by providing a drag-and-drop script construction GUI).

The deliberative subsystem is the primary locus of autonomous action selection in the cognitive controller (fig 2). This subsystem takes as input sensory data, child behaviour information, information on what step should be next executed from the therapy script, and higher-level direction from the wizard/self-monitoring subsystem. It then proposes what action should be taken next by the robot (this proposal is sent to the expression and actuation subsystem). In a normal script execution context, the deliberative subsystem is the primary driver of behaviour, which would typically propose the next script step.

There are however a number of circumstances in which this is not the most appropriate action to perform. For example, if the child is detected to have very low engagement with the task (as determined from the WP5 component/s, and/or information from WP4 sensory system saying the child is looking away for example), then it would be appropriate to attempt to re-engage the child with the robot/task prior to executing the next stage in the therapy script. In this case, the deliberative subsystem can choose to depart from the behaviour defined in the script, and instead propose a different behaviour.

5 Expression and Actuation Subsystem

The main functionality of this subsystem is to determine which combination of low-level actions the robot should execute next, and how these actions are to be performed. Suggestions for actions to take will come from three other subsystems: deliberative, reactive/attention, and self-monitoring, and the affective state generated by the deliberative subsystem, see left side of figure 4. Along with this, it is assumed that the supervising therapist, through the GUI, will determine (either beforehand or in real time) the aspects of robot behaviour that should be executed, from which relative priorities will be determined for the three subsystems. This covers for example whether external disturbances (a loud noise in the background, or the appearance of a new face) should be reacted to by the robot (by leaving the script for a while for example), or ignored (with the script rigidly adhered to). The Expression and Actuation subsystem will combine these sources of information in an appropriate manner, see Motion Mixer in figure 4, ensuring that the stability of the robot is maintained. For example, if a greeting wave is requested by the deliberative subsystem, and the reactive/attention subsystem wants to look at a face that has been detected, then the expression and actuation subsystem can combine the two by executing both (if the robot can remain stable by doing so). For a basic first step switches based on priority level could be used: i.e. if the script requests an action, execute it (and only it), but if there is no script action requested, then do what the reactive/attention subsystem proposes. However, the intention is to provide full behaviour mixing capabilities based on derived priorities from the therapists.

All this should be complemented by affective information, if this is available and appropriate to use. For example, the speed of motor execution could be related to arousal levels, or the choice of action sequence could be based on valence levels (if appropriate alternative sequences exist). This functionality will need to be switched on or off as required by the therapist based on child-specific considerations, and the relation to the therapy script (it may not appropriate to add emotional colouring to actions during the diagnosis procedure for example).

To approach such challenges, the first task should be to design a platform-independent representation of expressions. Different robots use the Facial Action Coding System (FACS) by Ekman and Friesen [2] to abstract away from the physical implementation of the robot face. FACS decomposes different human facial expressions in the activation of a series of Action Units (UA), which are the contraction or relaxation of one or more muscles. In a similar way, Body Action Units (BAU) will be defined together with a Body Action Coding System, where the different gestures are decomposed in the activation of BAUs. The BACS will point out the Action Units that need to be actuated for the generation of a desired gesture or body pose. This system avoids pre-programming of robot-dependent body poses and actions, which is relevant since humans are able to recognize actions and emotions from point light displays (so without body shape) [3].

The physical actuation of Action Units will depend on the morphology of the robot: a mapping will be needed between Action Units and physical actuators, this mapping will be specific to a robot platform and we will explore the possibility of learning this mapping. To translate this to the morphology of the robot, the Action Units need to be mapped to the degrees of freedom, and thus to the joints of the robot, see right side of figure 4.

A second task will be the categorisation of actions, comprised of temporal series of FACS and BACS, and the organisation in libraries that are accessible from the behaviour subsystems (Reactive, Attention and Deliberative). All actions for the different behaviours should be stored and expanded upon without the need to reprogram other subsystems.

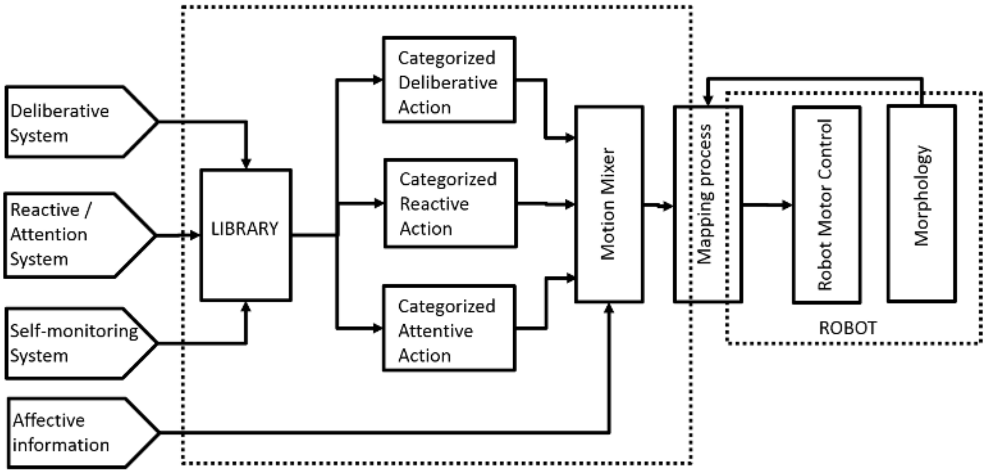

Figure 4: Overview of the Expression and Actuation subsystem. This subsystem receives inputs from several sources, categorizes them using the Library module and mixes them up to create a unique behavior. Such behavior is mapped into the joint configuration of the corresponding robot. This last process is done collaboratively between the subsystem and the robot.

6 Self-Monitoring Subsystem

The self-monitoring subsystem provides an oversight mechanism (or set of mechanisms) of the robot behaviour. It is intended to provide a check to prevent technical limits being exceeded (of the robot[3]), and to prevent any ethical boundaries being crossed. This subsystem should have some degree of autonomous behaviour, with the intention being that these checks be implemented in a set of predefined rules, with no role for learning within this subsystem.

During the discussions, it was proposed that the self-monitoring subsystem should also be integrated explicitly with the therapist GUI. In line with the principle of supervised autonomy established in the project, the therapist ("wizard") should be able to monitor the behaviour of the robot, and be able to intervene if necessary, either stopping the behaviour, modifying a behaviour, or setting an alternative behaviour. Having this oversight function go through the self-monitoring subsystem seems to be a reasonable solution. By specifying the required priorities for each subsystem depending on the needs of the therapy, and using the "alarm signals", the supervising therapist can stop the robot or modify its behaviour as desired.

Regarding both the autonomous oversight functions and the supervised actions, there are a number of issues that require exploration and further definition over the course of the project. One thing is how the robot should behave, and what feedback it should give to the child, should something go wrong. Possible alternatives are described in the DoW.

7 Action Primitives and Motor Execution

The behavioural functions of the action primitives required for completion of the therapy scripts have been defined. The execution of these is handled in a number of steps, as outlined in the "Robot

[3]This is mentioned here as it is listed in the DoW as a competence of the self-monitoring subsystem, however, this functionality is at least partially implemented in the low-level motor control system of the robot: see section 7.

Low-Level Motor Control" technical report. This provides an interface between the control system (handled in a Yarp-based system) and the API of the robot hardware (Naoqi in the case of the Nao). The purpose is both to provide a bridge between the two systems, and to provide information to behaviour planning and supervisory oversight regarding the progress of motor command execution, including why a fail occurs if it does. This can be used to inform future action selection for example (by providing feedback for learning).

In addition to this low-level control system, there is the possibility that hardware abstraction can be handled automatically: i.e. that motor commands at the joint level can be determined automatically for different robot embodiments, without having to manually encode each specific action.

8 Other aspects of the Cognitive Control System

8.1 User Models

One functionality that was not explicitly defined in the proposed architecture, WP6, or indeed elsewhere in the project, is some source of information on the child. This information could encompass personal identification and preference information that could be used in conversations (e.g. name, age, favourite colour, etc), and possibly also ASD diagnosis information (perhaps as emerging from the diagnosis interaction scripts).

These user models would enable, for example, inform learning mechanisms (within the deliberative subsystem for example) to link behaviours and outcomes with specific characteristics of individuals (indicated in figure 2). This information need only be uniquely identifiable rather than linked to a specific child - although the extent to which this can be done needs to be assessed in light of ethics considerations (cf. WP7 ethics manual draft, December 2014). Technically, in the first instance, a unique impersonal identifier may be used to represent an individual child. Where this information should reside, how it should be stored, etc, has not been decided. It would probably be useful however to coordinate this system with WP5, as the child behaviour interpretation methods may find such information useful too to be able to provide more personalised characterisations of engagement and performance for example.

References

[1] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, and B.ram Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, 2012.

[2] Ekman P and Friesen W. *Facial Action Coding System*. Consulting Psychologists Press, 1978.

[3] A. P. Atkinson, W. H. Dittrich, A. J. Gemmell, A. W. Young, et al. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception-London*, 33(6):717–746, 2004.

Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for Children with Autism Spectrum Disorders

TECHNICAL REPORT
Sandtray Wizard-of-Oz System for Turn-taking Intervention

Date: **23/03/2015**

Technical report lead partner: **Plymouth University**

Primary Author: **P. Baxter**

Revision: **1.0**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Service)	**PP**
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Summary

In this technical report we describe the software organisation of the Sandtray system created for the turn-taking diagnosis/intervention interactions. This system is based on the organisation defined by the WP3 software engineering standards, although at the moment does not fit into the rest of the DREAM system:this was to facilitate ease of setup and launch for the end-user (i.e. minimal installation, and no compilation required). The WoZ system provides a GUI from which the therapist can control the robot behaviour in the turn-taking task, and logs of the interaction are automatically stored for retrospective analysis.

Principal Contributors

The main authors of this document are as follows (in alphabetical order).

Paul Baxter, Plymouth University
Emmanuel Senft, Plymouth University

Revision History

Version 1.0 (P.B. 23-03-2015)
First version

1 Turn-taking Diagnosis/Intervention

The turn-taking diagnosis and intervention tasks are specified in D1.1. In the context of the Sandtray device, this is a task in which the child and the robot take turns in moving an image displayed on the touchscreen to one of two target locations. The robot plays the game with the child, and provides verbal indications of whose turn it is. Figure 1 shows the constructed Sandtray device. The task may be varied by changing the images to be sorted (e.g. characters, emotions, etc).

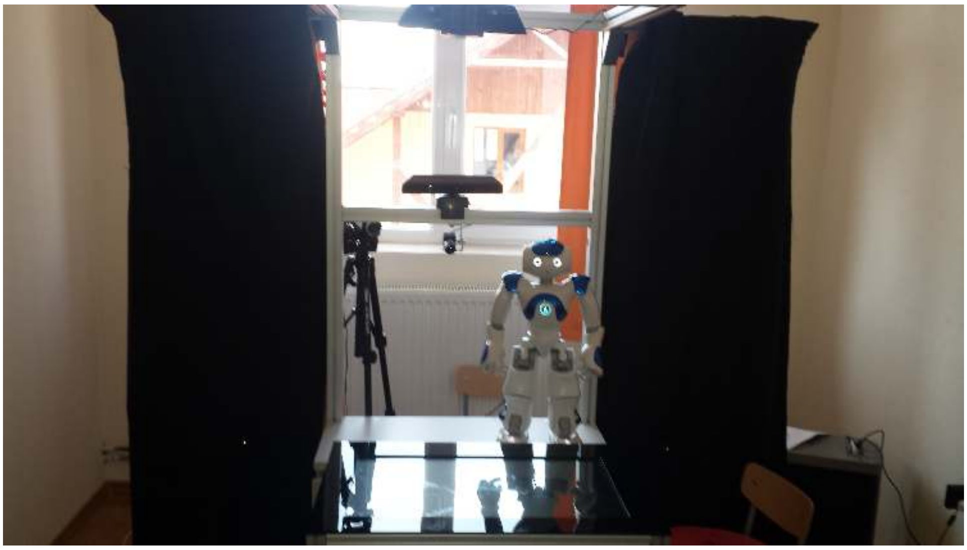

Figure 1: The final setup of the Sandtray with robot shown with the intervention table. The robot should ideally be centred in front of the screen to facilitate the interaction, and to ensure that the pointing behaviours are accurate.

In this first version of the task, a therapist provides all of the decisions regarding the behaviour of the robot: it is a full “Wizard-of-Oz” (WoZ) interaction. This enables experiments to take place before a full autonomous system has been implemented. Despite this, all of the components of the WoZ system are implemented using the software engineering standards established in WP3.

However, being designed as a standalone system with the intention of minimising the learning curve necessary for deployment and use, these components are not yet implemented in the context of the complete DREAM system architecture. The purpose of this technical report is to describe the system, its components and its use.

2 Wizard-of-Oz System

2.1 Dependencies

The intention is that this Sandtray WoZ system can be deployed easily and without need of recompilation on the target PC. There is only one dependency that requires installation before use: Yarp v2.3.63. This is needed in order to start the system. Instructions for installing this can be found on the project wiki (“Software Installation Guide”).

Additional setup instructions are as follows:

1. Copy folder ”dreamSandtray-release” into C: drive

2. Add ”C:/dreamSandtray-release/working/bin” to PATH (user) environment variable: this allows the precompiled executables to be found

3. Change the DREAM_ROOT environment variable to point to ”C:/dreamSandtray-release/”: if the full DREAM system is to be installed afterwards, be sure to change this back!

4. In file ”/working/config/config.ini”, change the IP address to that of the robot

5. In this same file, set the relative position of the robot to the touchscreen: this will be different depending on whether the robot is standing or crouching when playing the game; point of origin on robot is the neck joint

6. On sandtray machine, change the IP address to that of the Host PC (in ”settings.ini”)

2.2 Deployment

To launch the system, the following two steps are required:

1. On host PC: in directory ”/run” double click on ”run-sandtray.bat”: this will launch the system: if successful, the robot will readjust the position of its arms up and a GUI will appear; before the first launch, it may be useful to disable any firewalls, or else enable firewall access permissions for the four components that are launched; if launch fails, then close all windows and try again (if naoInterface.exe persists in failing to launch, then try restarting the robot)

2. After the robot has set its position and the GUI has appeared, then start the sandtray GameEngine: if successful, then will see confirmation on the terminal; if fails then check that the sandtray is on the same wireless network as the robot and the hot PC, and that the correct IP addressed have been set in the config files

3. Full interaction logs are automatically created: see section 2.4 below

2.3 Behaviour Control

Assuming a successful system launch, three terminal windows are spawned, and the control GUI (figure 2). This minimalistic GUI provides some feedback from the robot (success, or otherwise, of the desired motor commands - see low-level robot motor control technical report) and the Sandtray GameEngine (such as child successful or unsuccessful move, and the same for the robot, library change confirmation, etc).

The GUI provides two types of command. The first is presented in the middle column of buttons. These provide the verbal feedback that the robot can provide the child, such as a brief introduction, indication of whose turn it is, and feedback on the outcome of the categorisation. Additionally, a “rest” button is provided that moves the robot to a neutral position and switches off the motors. This provides a first level safety feature: if there is any chance of injury to child or damage to robot during movements, then this button will disable the robot. This is a basic feature at the moment, which will be extended in due course (see below).

The second type of command allows manipulation of the Sandtray GameEngine. “Good Move” and “Bad Move” make the robot perform the relevant classification of the image on screen (the image moves across the screen with the robot hand tracking it, thus providing the illusion of robot control). The “New lib” button presents a new image library on the touchscreen, and the “Reset lib” button presents the current image library on the screen again. “Shutdown” closes the GameEngine remotely.

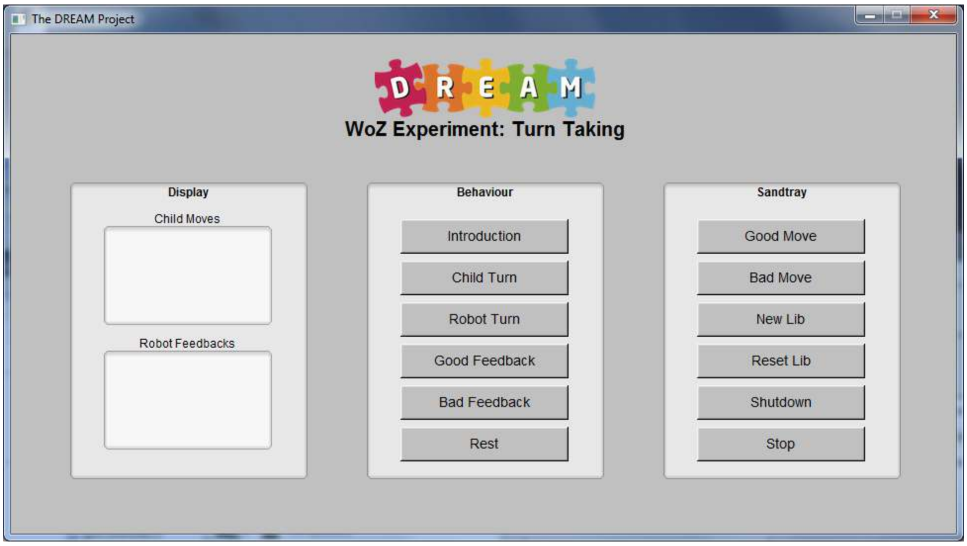

Figure 2: The Sandtray Turn-taking task control GUI. The windows on the left provide text feedback from the Sandtray and robot; the column of buttons in the middle provides robot verbal feedback; and the right-hand column of buttons provides robot behaviour control and GameEngine controls.

The "Stop" button is intended to provide a second level of safety control for the controlling therapist by enabling the interruption of robot movements that are currently under way. This functionality is under development, as part of the continued work on the low-level motor control (please refer to the relevant technical report).

2.4 Automated Interaction Logging

In addition to providing control, the Sandtray WoZ system also provides automated logging of all of the important events in the interaction that are detected by the Sandtray, or that are initiated by the therapist. These files are created automatically when the system is run: a filename with a timestamp is created in the directory where the batch file is located (where the system is launched from) - see section 2.2 for details.

This log file contains all events initiated through the GUI, events detected by the Sandtray, and motor command execution feedback. All events are logged with a timestamp (one second resolution), an identifier, and a description in comma-separated lines of a plain-text file. This facilitates parsing at a later time for retrospective analysis, replaying the interaction for diagnosis purposes, etc. This method of interaction logging will be extended as more data becomes available to the system.

3 Component Descriptions

The Sandtray Turn-taking WoZ system is comprised of four main components (in addition to the two external devices): sandtrayController, sandtrayServer, naoInterface and a GUI. sandtrayServer provides a communication interface between the yarp-based system and the Sandtray GameEngine. sandtrayController provides the primary coordinating role between the different components, containing some functionality that will soon be split between the Deliberative and Expression & Actuation sub-systems. The GUI provides the therapist-facing controls (described above). The naoInterface component

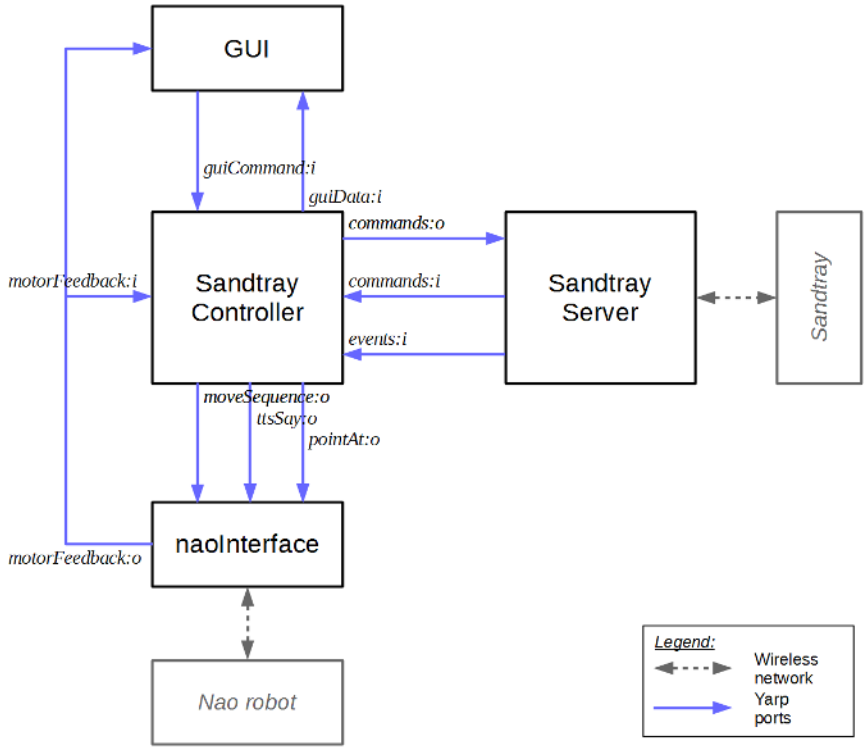

Figure 3: Components in the Sandtray WoZ control system. Communication between components is handled by yarp: shown are the component port names for the sandtrayController component (not all port names shown for clarity). Communication with the Sandtray device and Nao robot is handled over wireless networks outside of yarp.

3.1 gui component

This component is the interface used by the therapist to send request to the system. The current design is shown in figure 2. The description of the buttons may be found above.

The FLTK library is used to generate the GUI, the code is adapted from David Vernon's protoGUI, so the GUIutilities library is needed to generate it. The update is performed with a while(isStopping), could be improved with a rated thread or something else. The organisation of files is the same as described in YarpGenerator tech report: the 4 files needed for yarp, a yarpInterface class, a controller class where the main code is, and the display class containing all the information required for designing and running the GUI. Currently each button in display is linked to a static callback redirecting to a callback in display.cpp, which calls the appropriate function in controller.cpp, this could probably be simpler.

This component also uses the childName variable in the main config.ini to generate the string send to load the introduction behaviour to make it personal to each child. So one introduction behaviour need to be exported for each different child and should use the following convention: introduction-childName , e.g. "introductionGeorges".

3.2 sandtrayController component

The sandtrayController component is the main and central component. This is the one managing the interaction between the other components. It has input and output ports from/to all the other components. It gets the command from the gui and relay them to the server or the naoInterface and act in accord with the feedback obtained.

As this is the central component, the logging is done from here. The file structure used is the classical YarpGenerated one, with an additional logger class (.cpp and .h) to be able to generate logging files. Currently the logging is disabled but the file is still created, this could be changed.

This file does some string processing, so new methods have been added to handle that. It also integrate a Link class allowing to Link an action to a previous if a certain state is reached. The linking process is described in depth below.

This component is also responsible of the bezier calculation and the transformation from pixel coordinates to robot's ones. All the variables are defined in sandtrayControllerController.h except the horizontal and vertical distances between the robot and the screen which are defined in the global config.ini file in centimetre.

3.3 sandtrayServer component

This component is responsible of the interface between Yarp and the Sandtray, its main functionality is to transform Yarp message into sockets one and vice versa.

The global structure is the same as the others components. In addition, two class are define in the controller file: SandtrayControl and SandtrayEvent. The first one is a classical class which handle the communication along the *command socket port* with the Sandtray, which is used to transmit information from the controller to the Sandtray, and receive the answers. The communication is only triggered by the SandtrayController, and handled by Yarp callbacks, so no while loop is required.

The other one manges the communication along the *event socket port* and in that case, the discussion is triggered by the Sandtray. In the current implementation, this class wait using a while(isStopping) to get information from the Sandtray, so yarp::os::Thread is inherited from.

A limit of the current implementation is that the Sandtray needs to be started after the server component.

3.4 naoInterface component

This component implements the action primitives as defined in the deliverables. The idea is to process commands to the robot, check if the command is possible, avoid conflicts and provide a way to stack commands.

Currently the commands allowed are: pointAt, say, execute a behaviour. In addition to the classical files, an action class has been added, which is used to serialise the different type of actions: movement, text to speech or behaviour. All the tests performed are implemented here. The other new class is naoInterfaceModule, this provide the interface with Nao, with all the functions robot specifics. In principle the code should be easily modified to adapt to another robot, only this class and the refFeature map used to avoid conflicts should be modified. More information is available in the Low Level Control tech report.

Comparing to the previous implementation, multiple features have been added, and few changes have been implemented in the flow, no completion check is performed when an action is received. Multiple action cannot be successfully check anymore, like tts or behaviour and the movements can be composed of multiple points, so a previous completion check makes no sense. The ISTARTED message has been also overloaded, now it can be followed by a " sub id" with id a negative number used to synchronyse the controller and the naoInterface.

The component now integrate a pointAt function which does a simple inverse kinematic to transform x, y, z coordinate in joint space. It selects first an arm depending on how many points are more on the right side or the left one, then compute the joint angles. This function can take multiple points with the parameter x, y, z and t, and only a subset of the movement can be executed.

3.5 Detailed description

3.5.1 Action linking and synchronisation between the naoInterface and sandtrayController

In the previous implementation, each action required by a component and executed by the naoInterface was identified by a unique id. However this id was provided by the naoInterface and when a component required an action, it had no way to know the id of this exact action, multiple components can require action at the same time. We wanted a way to serialise two or more action. To do so, when a component request an action it needs to know what is this action's id to be able to command a new action when this one is finished depending on its result.

To allow this synchronisation, a system of subId was implemented. When a action is requested by a port, it can be overloaded by adding a new parameter at the end of the parameter list to assign a subId to an action. The subId is a negative int allowing to know what is the id (in naoInterface) assigned to a defined action. In naoInterface, this subId is extracted and added to the action and when the action is started, the subId is added at the end of the ISSTARTED message to allow the other component to know what is the id of the action with a specific subId. Currently, this subId is displayed and used only with the ISSTARTED message. In the future it should be important to use it also in case of direct failure.

In our setup, only the SandtrayController requires the use of synchronised actions. This is done via the Link class. Defined in sandtrayControllerController.h, this class define an action: id and subId, a linking condition ($\_status$), a consequence and a pointer to another link if needed. This allow us to perform a specific action when a defined action is finished, and cascade it if required. Currently, after the completion of and action, we can request a new tts, movement or behaviour, the type of action is stored in $\_consequence$, the parameters of this action are stored in the two strings $\_param1$ and $\_param2$ or $\_move$.

To create a Link, we need to know what is the subId the causal action, the one which will trigger the Link, later, this subId will be associated to a proper id (the same as used by naoInterface). If we want to chain multiple action, we need first to define all our actions with the proper subId (defined in the port parameters), and then create the different Links with the subId assigned to their causal action and giving them the parameters for the consequence action. Then, each consequent Link can be added to the previous Link in $\_nextLink$, and the first Link can be added to the $\_currentLink$ vector in the controller class.

A subId variable is stored in the controller, and is decremented once for each Link, and this is used to assign subId to an action.

Currently, when an action is a success, the $\_nextLink$ is added to the $\_currentLink$ vector, and if it is a faillure, it is destroyed. However there is a memory leak, if we have a chain of links, and the first one is failed, only the next one is deleted, not all the chain...

3.5.2 Flow when move requested

The flow is as defined in the previous the design. The added part is the following: the sandtray returns coordinates for the bezier move in pixel, this is relayed to the controller via the server. Then the controller extracts the bezier points and transform the coordinate in robot space.

A first pointAt is prepared with all the bezier points plus a one at the end of the vector to signify that only the first movement should be perform. The current subId is added also at the end of the vector. Then a link is create with this subId and the success status to execute the full movement and the boolean $\_synchronizedSandtray$ is set to 1 meaning that the command need to be sent to the Sandtray also. The subId variable is decremented and added to the movement after a 5, meaning that we want to perform the full movement with the new subId. A last link is created with this subId and the behaviour *init* in parameter, to reset the robot. This last link (behaviour) is added in the $\_nextLink$ pointer of the movement Link, which is added in $\_currentLink$.

Finally the first movement (initial pointAt) is sent to naoInterface. There the subId is extracted, and sent with the ISSTARTED message to the controller, which assign this id to the first link. When the pointing is completed, the first link is triggered: we send to the naoInterface and the Sandtray the request to move and we add the second link to the $\_currentLinks$. Once the full movement is received by the naoInterface, it send to the controller a new ISSTARTED message with the new id and subId. Similarly this is handled by the controller to assign the right id to the second link. And finally when the action is completed, the last link is triggered to reset the robot.

4 Sandtray GameEngine Management

4.1 Sandtray GameEngine

The Sandtray sorting task software is already loaded onto the touchscreen, with some sample image libraries (see next section for details of these). There is an executable in a folder on the desktop, and a shortcut on the desktop itself. Double-clicking this starts the programme in full-screen mode. This automatically loads all available image libraries, and cycles through them in numerical/alphabetical order.

There is only one keyboard shortcut that is required for operating the sorting game: the ESC key exits the programme and returns to the desktop. We advise not to do this in front of the child unless absolutely required - we recommend having the sorting game on the screen when the child enters the room.

Further to this there are two on-screen buttons that can be used to control the sorting game. The circular arrow leads to a reset of the current library (i.e. the same images are displayed again). The other icon (a sun with two + symbols...) indicates a switch to the next library. With further integration with the rest of the system, switching to specific image libraries will be possible at a later date. These on-screen buttons can be disabled if required in the config file (see below).

The software has a number of options that can be modified in a configuration file, located in "settings.ini". There should not be any need to change the paths in this file. The "robotiP" field should be set to the IP address of the host PC being used. The "game" options control aspects of the GUI, as described below (table 1).

Table 1: Sandtray sorting game options, can be found in settings.ini

Setting	Description	Default
LadderRungs	The number of fields visible for classified images	5
LadderWidth	Width of fields (in pixels)	50
ReserveTests	(do not modify) deprecated functionality	false
TestLibStart	(do not modify) deprecated functionality	-1
UseButtons	Whether to display control buttons on screen	true
ShowFeedback	Whether to display feedback on classifications	true
OneAtATime	Display images one at a time on screen	true
CentreImages	Display images in the centre of screen (random otherwise)	true

4.2 Sandtray Image Library Management

The Sandtray sorting game is based on *image libraries*. Each image library has a unique identifier, and contains a set of images (in .png format). Each library is in a separate folder in /images/libraries/. The name of each library folder follows a specific format:

libxx_name_opt

Where *xx* is an integer (e.g. *01*), *name* is a string identifier (e.g. *carbohydrates*), and *opt* is an optional extension that provides additional information.

For the purposes of the present experiments (turn-taking), a two category sorting task is assumed (one, two of four category sorting tasks are possible). Each library is a folder containing a number of images. There are no explicit limitations on the number of images per library, although resource issues (all images are pre-loaded at run-time) may mean that splitting up large libraries into parts is necessary. Two category images must be defined. These can be any image of the same type and size as standard images, but with the following naming convention:

catx_name.png

Where *x* is either *A* or *B*, and *name* is a string identifier of the category. For example, the following two filenames may be used to define two categories: *catA_high.png* or *catB_low.png*.

Standard images (in .png format) have to be assigned to one of the two categories (A or B), and their filenames should follow a similar but slightly different naming convention:

xNN_name.png

Where *x* is the category to which the image belongs (as defined for the category images above), *NN* is a unique image integer identifier (e.g. *01*), and *name* is a string identifier that provides some descriptive information. Examples of suitable image file names are *A01_chicken.png* or *C11_lasagne.png*.

Please see the example image libraries supplied with the Sandtray for indications of appropriate image size and resolution (as tested with hundreds of UK primary school children in the approximate age range of six to nine): please ensure that the images are not too large, as this will cause problems.

The string identifiers (and optional extra information) attached to the names of the libraries and the images are not used at the moment. However, they will be used to enable the robot to refer to specific objects, and specific properties of those objects as the autonomous system is developed. It is therefore worth adding this information to the folder and file names during the creation process.

Project No. 611391

DREAM
Development of Robot-enhanced Therapy for
Children with Autism Spectrum Disorders

TECHNICAL REPORT
Manual for the use of Choregraphe boxes in Wizard of Oz experiments

Date: **11/02/2015**

Technical report lead partner: **Vrije Universiteit Brussel**

Primary Author: **Pablo Gómez**

Revision: **1.0**

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Service)	**PP**
RE	Restricted to a group specified by the consortium (including the Commission Service)	
CO	Confidential, only for members of the consortium (including the Commission Service)	

Contents

Summary

The purpose of this manual is to help UBB in the development of the Wizard of Oz experiments within Work Package 2. Both PLYM and VUB have collaborated to develop the corresponding modules in Choregraphe. This manual aims at being a reference point to ease the habituation of the therapists to the software.

Principal Contributors

The main authors of this document are as follows (in alphabetical order).

Pablo Gómez, Vrije Universiteit Brussel

Revision History

Version 1.0 (P.G. 11-02-2015)
First draft

1 How to use this manual

In order to make the software the more understandable as possible we have divided each of the scenarios in Deliverable D1.1 into different project files. So that, in between breaks, the forthcoming scenario has to be loaded. The procedure to follow to load a project will be explained in Section 2. Once opened, each project file includes a set of boxes to make the robot perform an action or say something. Each of these boxes is identified by a self-explaining name. To make easier the management of the boxes during the experiment we have not grouped them chronologically but into families of similar behaviors, i.e. all text boxes are close to each other, actions regarding pointing are together, and so on. In addition, a representative image identifies each family of behaviors.

At the end of the manual some notes about the limitations of this software are included. We must not forget that it is following a Wizard of Oz methodology so most of the work to follow the script is required from the therapists.

2 How the software is organized

As mentioned above each of the scenarios in the deliverable D1.1 is one project file named after the name included in D1.1.

2.1 How to load a project file

In order to load a project file, once Choregraphe has been opened, click on *File*, *Open project...* and select the one you prefer from the list of projects, and click *Open*. Then, you will see something like in Figure 1.

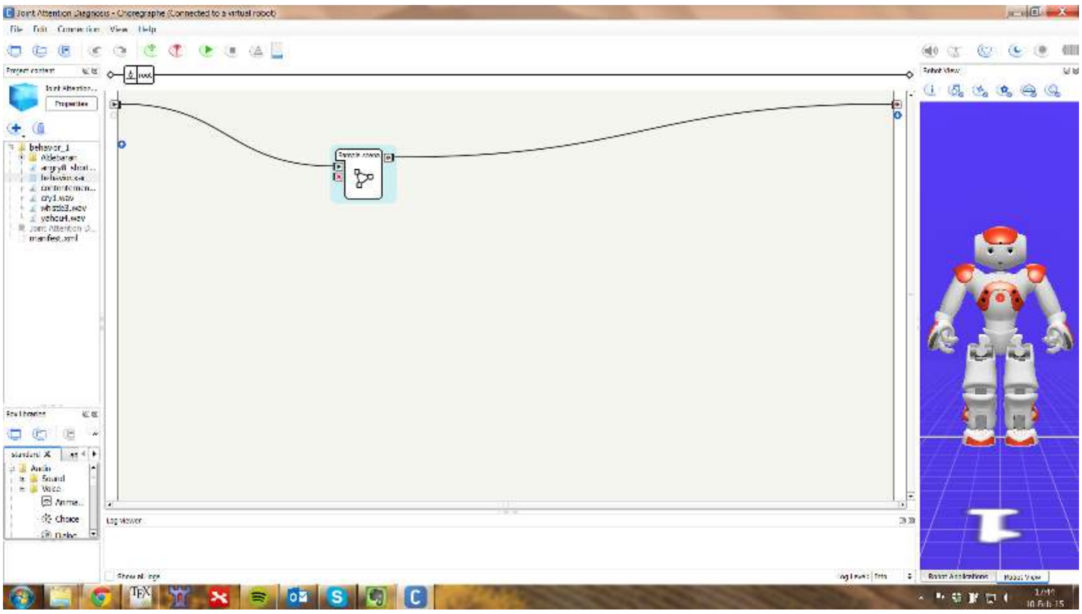

Figure 1: Opening a project file.

By double-clicking on the box that appears, Figure 2 will show up. You need to click on the *Interaction* block shown in blue. All the required behaviors for the corresponding scenario will be shown.

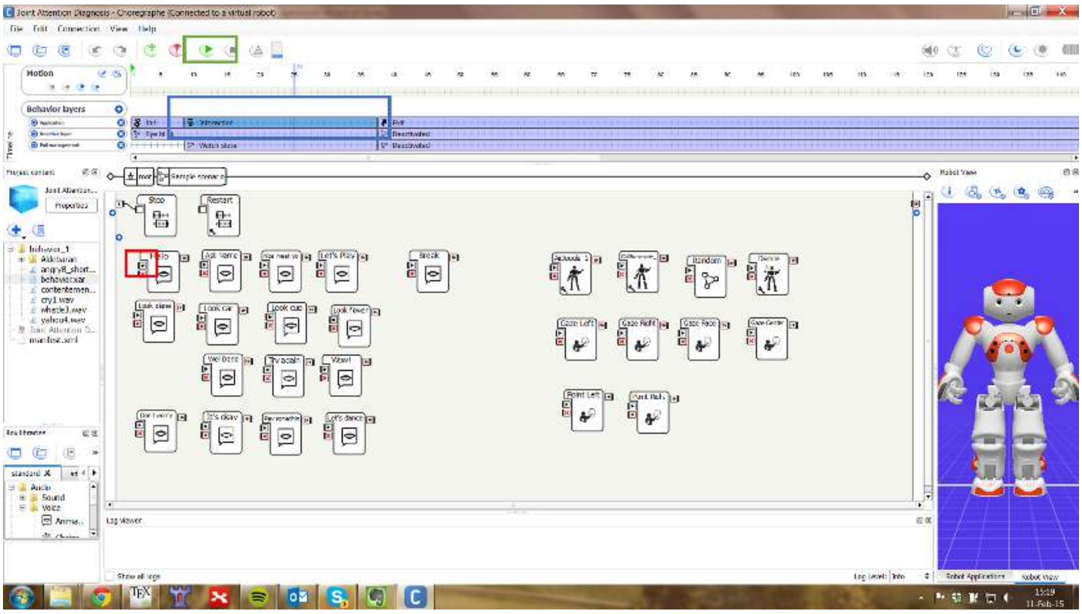

Figure 2: Accessing the behaviors within a project file. To activate each behavior click twice on the small *Play* button on the left of each of them.

All behaviors include a name to briefly identify what they do. They are grouped according to their functionality. There is no chronological order.

We have included a *Restart* button to use in case something went wrong to initialize all the variables and start the intervention over. There is a *Stop* button which is required by Choregraphe, you don't need to pay attention to it.

2.2 How to run a project file and stop it

In order to run a project file you just need to click on the *Play* button, in green in Figure 2. Once it is running, you can activate each behavior double-clicking on the small *Play* button each behavior includes, in red in Figure 2. When such behavior is activated the button changes its color to green for a few seconds. If it didn't happen the behavior was not correctly activated.

In order to stop a project file, is as easy as clicking on the *Stop* button next to the *Play* button within the upper menu.

3 Description of the scenarios

In this Section, the scenarios developed will be briefly described. All scenarios include the required behaviors in Sections 3.1 and 3.9 in Deliverable D1.1. Those behaviors are described here in Subsection 3.1.

3.1 Actions at the start of all RET tasks

At the beginning the robot stays standing up in a relax pose waiting to start. Two motivating motions are provided to try to engage the child, see *AirJuggle* and *CallSomeone* boxes in Figure 3 as well as a box to make the robot dance, see blue frame in Figure 3. There is also a box called *Random* which creates 6 different behaviors with the purpose of engaging the child.

There are several text boxes to start the session with the child and one to suggest to have a break. Click on each of them to make the robot say something. These boxes are those under the red frame in Figure 3. There are also text boxes to re-engage the child if something was unexpected, see yellow frame in Figure 3

3.2 Joint Attention Diagnosis ADOS

Within this project file there are text boxes to be used to make the robot say *Look at that <object>* where object could be a plane, a car, a cup or a flower, see purple frame in Figure 3; text boxes to provide feedback to the child after each interaction; and there are also boxes to direct the gaze of the robot (left, right and center) and boxes for pointing to left and right see green frame in Figure 3.

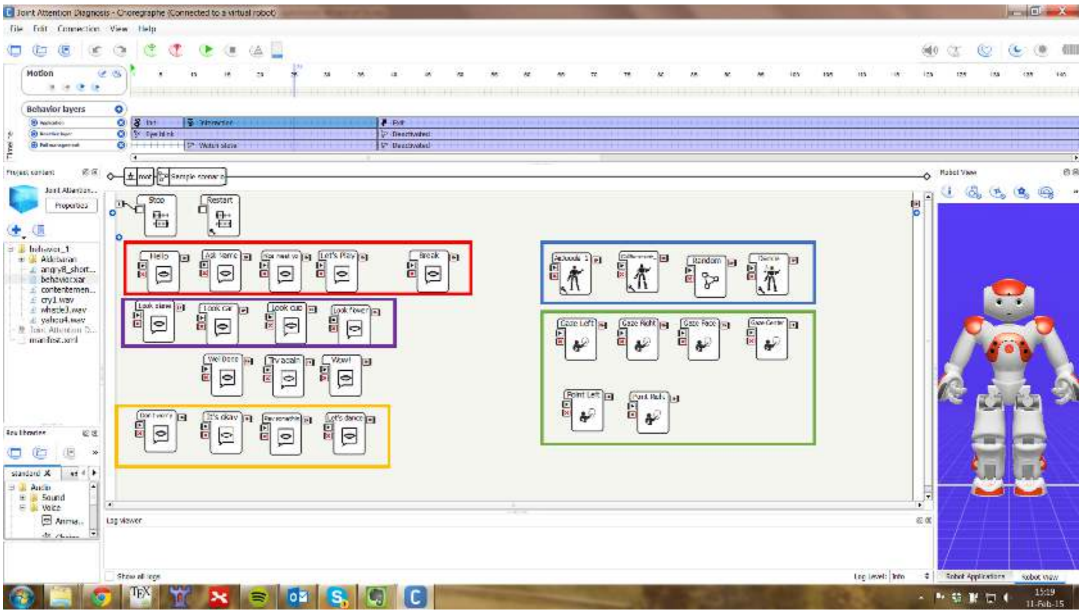

Figure 3: Joint Attention Diagnosis project file.

3.3 Joint Attention Intervention

In addition to what was included in the previous scenario, within this project file there are text boxes to ask the child to choose an emotion, purple frame in Figure 4; and boxes for expressing different emotions, see green frame in Figure 4.

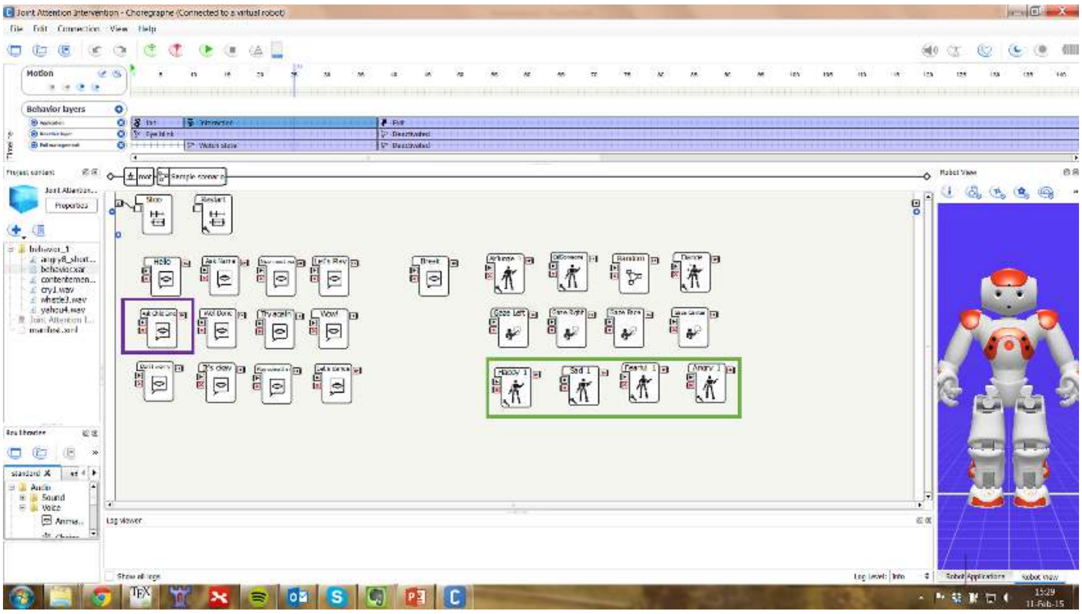

Figure 4: Joint Attention Intervention project file.

3.4 Imitation Diagnosis with Objects

Additionally to what there are in other project files, within this one there are text boxes to ask the therapist to give the robot certain object, see purple frame in Figure 5; a box to ask the child to replicate the motion, in yellow in Figure 5; and boxes for making different gestures with and without the objects, see green frame in Figure 5.

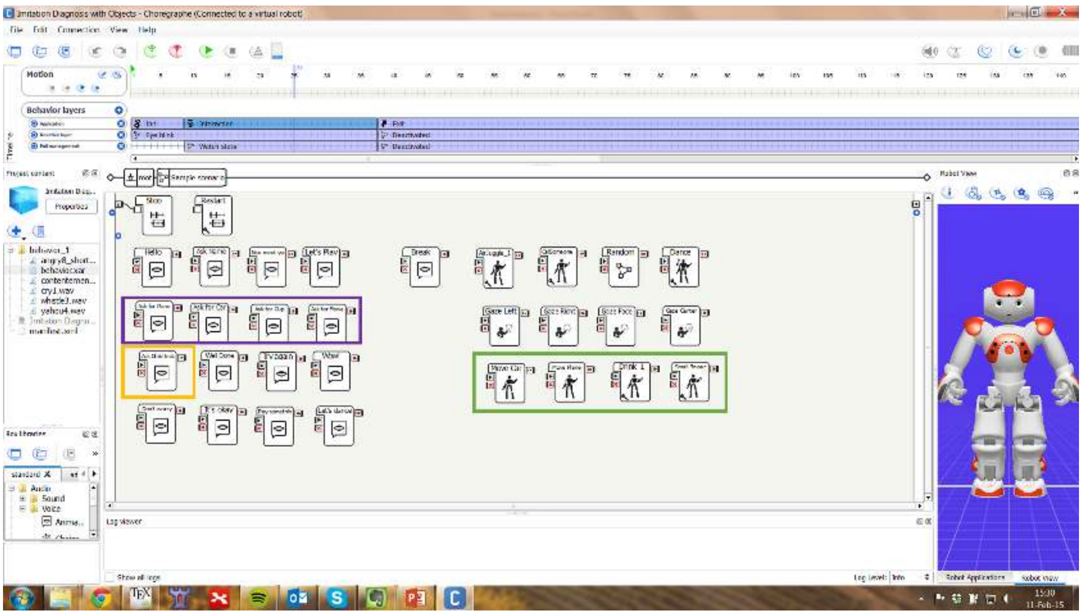

Figure 5: Imitation Diagnosis with Objects project file.

In order to make the robot pick up an object, the therapist should approach the object to the robot and touch its head, then the robot will open its hands. With a second touch in the head, the robot will close the hands grabbing the object. To make the robot drop the object, the same procedure should be followed.

3.5 Imitation Diagnosis without Objects

For this project file there are specific boxes to make the robot make 4 different motions (cover its eyes, touch its head, airplane arms and wave with one hand), see green frame in Figure 6.

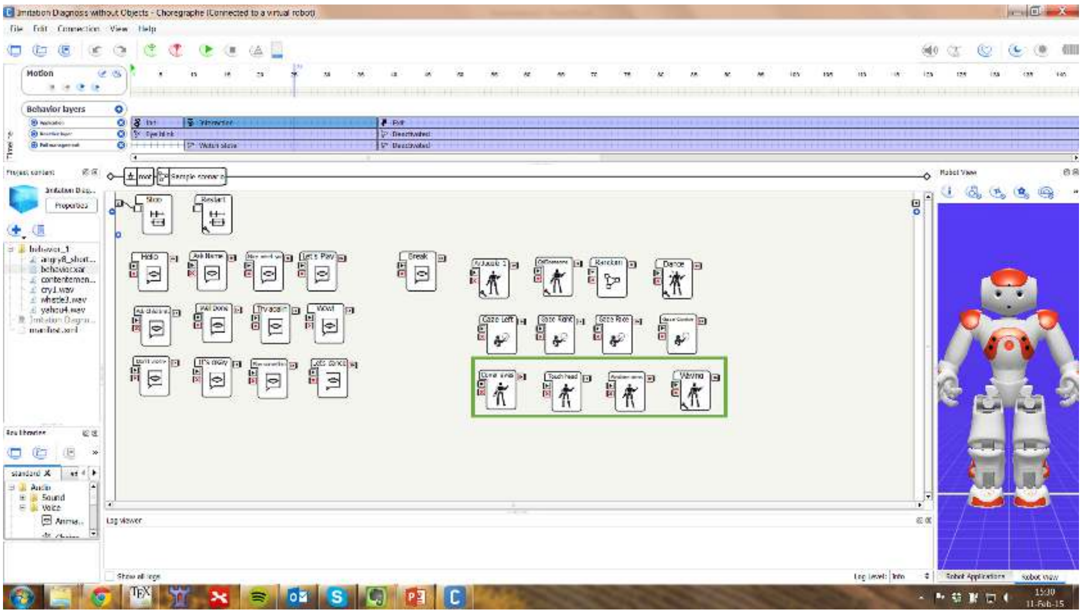

Figure 6: Imitation Diagnosis without Objects project file.

3.6 Imitation Intervention without Objects

This project file includes text boxes already described in previous scenarios.

3.7 Turn-taking diagnosis

The turn-taking diagnosis behaviour for the WoZ experiments is handled by a separate system based on the software engineering framework established in WP3. Please refer to the technical report "*Sandtray Wizard-of-Oz System for Turn-taking Intervention*" for more details.

3.8 Turn-taking intervention

The turn-taking intervention behaviour for the WoZ experiments is handled by a separate system based on the software engineering framework established in WP3. Please refer to the technical report "*Sandtray Wizard-of-Oz System for Turn-taking Intervention*" for more details.

4 Limitations of the software

As the experiments will follow the Wizard of Oz methodology, the therapists are responsible for following the script of each scenario. There is no autonomous behavior within this software at all. Moreover, in case it is needed more boxes can be added.

If you need further information about Choregraphe, you may find it in Aldebaran documentation (`http://doc.aldebaran.com/2-1/news/2.0/choregraphe_rn2.0.html`).

5 Miscelanea

The software has been developed using Naoqi version 2.1.2.17. If you need to upgrade it, please follow the instructions from the wiki (`https://dreamproject.aldebaran.com/projects/dream/wiki/Nao_software`).

Period 2

Touchscreen-Mediated Child-Robot Interactions Applied to ASD Therapy*

Paul Baxter[1], Silviu Matu[2], Emmanuel Senft[1], Cristina Costescu[2], James Kennedy[1], Daniel David[2] and Tony Belpaeme[1]

[1]*Centre for Robotics and Neural Systems, The Cognition Institute, Plymouth University, U.K.*

[2]*Department of Clinical Psychology and Psychotherapy, Babes-Bolyai University, Romania*

Abstract. Robots are finding increasing application in the domain of ASD therapy as they provide a number of advantageous properties such as replicability and controllable expressivity. In this abstract we introduce a role for touchscreens that act as mediating devices in therapeutic robot-child interactions. Informed by extensive work with neurotypical children in educational contexts, an initial study using a touchscreen mediator in support of robot-assisted ASD therapy was conducted to examine the feasibility of this approach, in so doing demonstrating how this application provides a number of technical and potentially therapeutic advantages.

Keywords: ASD, Robot-Assisted Therapy, Sandtray

INTRODUCTION

The application of robots to aid in the therapy of children with Autistic Spectrum Disorders (ASD) has become increasingly established [1], [2], with evidence suggesting that it can provide beneficial outcomes for the children [3]. In addition to this, recent efforts have emphasised providing an increasing degree of autonomy for the robot [4].

Providing such autonomous behaviour in interaction contexts is a challenging task, with sensory and motor limitations imposing a number of constraints. In our previous work, we have developed a methodology that makes use of a touchscreen mediator between children and robots to overcome a number of these difficulties: the Sandtray [5]. In this setup, a child and a robot engage in a collaborative task that is provided on the touchscreen (e.g. sorting of images into categories). The Sandtray has been successfully applied to a range of neurotypical child-robot interaction studies in various contexts, for example behavioural alignment [6], education [7], and others. As the Sandtray was inspired by the therapeutic intervention of sandplay (with this having proposed advantages for children with ASD [8]), we now seek to apply this same methodology to robot-assisted ASD therapy.

Touchscreens (without the robot) have found previous applications to this domain [9]. For example, a touchscreen has been used to enforce collaborative activity between pairs of children with ASD, resulting in an increase in coordination and negotiation behaviours [10], a finding supported elsewhere [11]. Furthermore, there have been attempts to enable sandplay therapy-like interactions with touchscreens [12],

*This work was supported by the EU FP7 project DREAM (grant number 611391, http://dream2020.eu/).

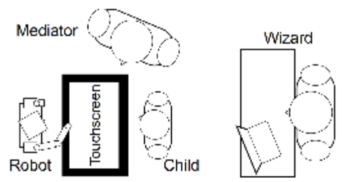

Fig. 1. Indicative setup and use of touchscreen for child-robot therapeutic interaction - robot is controlled by a wizard, and the mediator provides input to the interaction if needed (*not to scale; positions are indicative only*).

although our approach differs in both application context and involvement of the robot. These studies indicate the suitability of using touchscreens for children with ASD.

There are a number of advantages afforded by the use of such a mediating touchscreen in HRI. Firstly, it provides a shared space for collaboration that does not require complex manual dexterity for either the child or the robot; indeed it provides the same affordances for both interactants (pointing and dragging). Secondly, it reduces the sensory processing load (vision processing) on the robot since information on screen-oriented activity by the child can be obtained directly from the touchscreen. Thirdly, it provides a straightforward means of changing the task (or more broadly the interaction context) by just changing the images displayed on the screen: for instance, a sorting task can be appropriate for domains as diverse as mathematics and nutrition just by changing the pictures displayed.

The aim of this contribution is to motivate and illustrate how such touchscreen mediators can specifically serve as useful tools in the domain of robot-assisted therapy by first describing an application currently in progress, and then discussing the opportunities and challenges for the future.

APPLICATION CASE STUDY: TURN-TAKING

An initial application to ASD therapy has been implemented and evaluated. Turn-taking is an important social skill that is used as part of therapeutic interventions [13]. We have created an emotion image categorisation task (using *sad* and *happy* faces) on the Sandtray for a child and Nao robot to play, with robot verbal behaviour used to encourage turn-taking behaviours. For this study, the robot was explicitly remote controlled (*wizarded*) by a remote operator (fig. 1).

With a four year-old girl with ASD, six interaction sessions with the Robot-Sandtray turn-taking task were con-

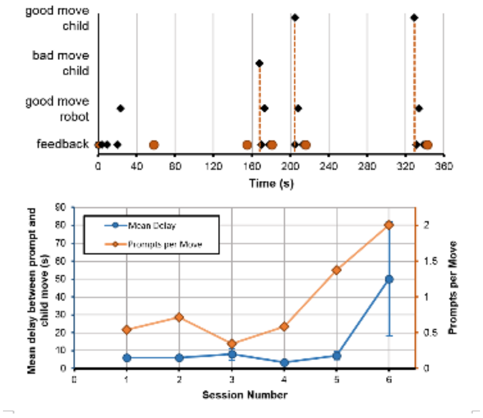

Fig. 2. (*Top*) Sample data from the sixth child-robot Sandtray turn-taking interaction session. The *feedback* was employed to encourage the child to move and to give them feedback. Orange circles indicate robot encouragements for the child to take a turn. (*Bottom*) Trends over six sessions, showing change in delay between robot prompt and the child moving, and the mean number of prompts per child move (*with 95% CI*).

ducted over a period of four weeks. Other robot-based therapy activities were conducted at a separate time. Each interaction had a mean length of 11:06 mins (*sd 5:03 mins*).

Since interaction data can be captured through the touchscreen, it is possible to retrospectively examine the events that occurred and their timing. Considering the relationship between robot encouragement and child moves in a single interaction (e.g. fig. 2, top), the data suggest that both the number of robot encouragement instances required before the child made a move, and the delay between suggestions and actual moves increases over time (fig. 2, bottom). A clinical explanation for this relationship is not proposed here, although the ideal behaviour in this context is a turn-taking interaction with the robot, without necessarily requiring explicit prompting. What can be noted though is that data such as these provide some insight into the interaction between the child and the robot over time.

DISCUSSION AND OPEN QUESTIONS

The examination and use of touchscreen-derived information has two benefits. Firstly, it may come to constitute an additional source of information for the therapist to aid in diagnosis or inform future therapy, with additional processing making aspects of emotion available for example [14]. The extent to which this is clinically useful is an open question that requires investigation. It should however be noted that we do not suggest that such data can replace traditional diagnosis information, rather that it can provide supplemental information. It should be further noted that the touchscreen-derived information alone is likely to be insufficient to provide a complete characterisation of the child's behaviour.

Secondly, since the information captured by the touchscreen is directly accessible to the robot system, it can be used by the robot to adapt its behaviour to the specific circumstances of an individual child in individual interactions, e.g. [6]. In the case of autonomous robot behaviour, such a source of information that does not require the overhead of complex visual or audio processing is a significant benefit.

Extensive previous work has been conducted with this touchscreen mediated interaction between (neurotypical) children, and robots. While this has shown that the touchscreen effectively constrains the content of the interaction (thus facilitating robot autonomous behaviour) [15], it is an open question as to whether a similar effect (such as helping to maintain focus on the interaction) is observable for children with ASD, or over what time scales such an effect may be manifested.

To conclude, we have presented data from an example set of interactions between a child with ASD and a robot in the context of the Sandtray. This provides an illustration of the type of data that is readily available through the use of the touchscreen mediation technology. While further development and data collection is required (and is ongoing), we suggest that the use of touchscreens as mediators for child-robot interactions in the context of ASD therapy provides benefits in terms of behaviour characterisation and technical feasibility that should be further taken advantage of.

REFERENCES

1. B. Robins et al, "Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills?" *Universal Access in the Information Society*, 4(2): 105–120, 2005.
2. B. Scassellati et al, "Robots for use in autism research," *Annual review of biomedical engineering*, 14: 275–94, 2012.
3. C. A. Costescu et al, "The Effects of Robot-Enhanced Psychotherapy: A Meta-Analysis," *Review of General Psychology*, 18(2): 127–136, 2014.
4. S. Thill et al, "Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook," *Paladyn*, 3(4): 209–217, 2013.
5. P. Baxter et al, "A Touchscreen-Based "Sandtray" to Facilitate, Mediate and Contextualise Human-Robot Social Interaction," in *7th HRI Conference*. Boston, MA, U.S.A.: IEEE Press, 2012, pp. 105–106.
6. P. Baxter et al, "Cognitive architecture for human-robot interaction: Towards behavioural alignment," *Biologically Inspired Cognitive Architectures*, 6: 30–39, 2013.
7. J. Kennedy et al, "The Robot Who Tried Too Hard: Social Behaviour of a Robot Tutor Can Negatively Affect Child Learning," in *10th HRI Conference*. Portland, Oregon, USA: ACM Press, 2015, pp. 67–74.
8. L. Lu et al, "Stimulating creative play in children with autism through sandplay," *Arts in Psychotherapy*, 37: 56–64, 2010.
9. W. Chen, "Multitouch Tabletop Technology for People with Autism Spectrum Disorder: A Review of the Literature," *Procedia Computer Science*, 14(1877): 198–207, 2012.
10. A. Battocchi et al, "Collaborative puzzle game: a tabletop interface for fostering collaborative skills in children with autism spectrum disorders," *Journal of Assistive Technologies*, 4(1): 4–13, 2010.
11. G. F. Mireya Silva et al, "Exploring collaboration patterns in a multitouch game to encourage social interaction and collaboration among users with autism spectrum disorder," *Computer Supported Cooperative Work (CSCW)*, 24(2-3): 149–175, 2015.
12. M. Hancock et al, "Supporting sandtray therapy on an interactive tabletop," *28th CHI Conference*, pp. 21–33, 2010.
13. C. A. Pop et al, "Enhancing play skills, engagement and social skills in a play task in ASD children by using robot-based interventions. a pilot study," *Interaction Studies*, 15(2): 292–320, 2014.
14. Y. Gao et al, "What Does Touch Tell Us about Emotions in Touchscreen-Based Gameplay?" *ACM Transactions on Computer-Human Interaction*, 19(4): 1–30, 2012.
15. J. Kennedy et al, "Constraining Content in Mediated Unstructured Social Interactions: Studies in the Wild," in *5th AFFINE Workshop at ACII 2013*. Geneva, Switzerland: IEEE Press, 2013, pp. 728–733.

Human-Guided Learning of Social Action Selection for Robot-Assisted Therapy

Emmanuel Senft, Paul Baxter, Tony Belpaeme
Centre for Robotics and Neural Systems,
Cognition Institute Plymouth University, U.K.
{emmanuel.senft, paul.baxter, tony.belpaeme}@plymouth.ac.uk

Abstract

This paper presents a method for progressively increasing autonomous action selection capabilities in sensitive environments, where random exploration-based learning is not desirable, using guidance provided by a human supervisor. We describe the global framework and a simulation case study based on a scenario in Robot Assisted Therapy for children with Autism Spectrum Disorder. This simulation illustrates the functional features of our proposed approach, and demonstrates how a system following these principles adapts to different interaction contexts while maintaining an appropriate behaviour for the system at all times.

1 Introduction

Humans are interacting increasingly with machines, and robots will be progressively more important partners in the coming years. Human-human interactions involve high dimensionality signals and require complex processing: this results in a large quantity of data that ideally needs to be processed by an autonomous robot. One potential solution is the application of machine learning techniques. Specifically, online learning is desirable, however, some level of initial knowledge and competencies are required to avoid pitfalls in the early phases of the learning process, particularly in contexts where random exploration could lead to undesirable consequences.

In this paper, we propose an approach inspired by, but separate from, learning from demonstration to guide this early learning of an action selection mechanism for autonomous robot interaction with a human, by taking advantage of the expert knowledge of a third-party human supervisor to prevent the robot from exploring in an inappropriate manner. We first present the formal framework in which our action selection strategy learning takes place (section 2), then illustrate this with a case study from the domain of Robot Assisted Therapy for children with Autism Spectrum Disorder (ASD), where the incorrect selection of actions can lead to an unacceptable impact on the goals of the interaction (section 3).

Appearing in Proceedings of the 4^{th} Workshop on Machine Learning for Interactive Systems (MLIS) 2015, Lille, France. JMLR: W&CP volume 40. Copyright 2015 by the authors.

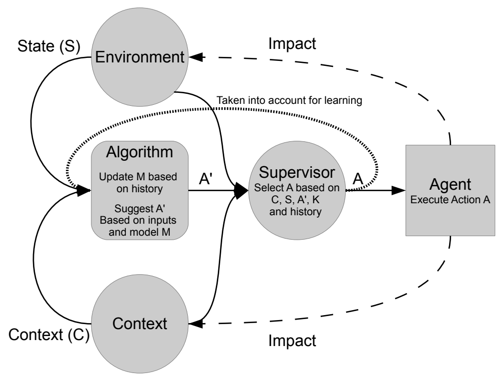

Figure 1: The supervised online learning of autonomous action selection mechanism.

2 Supervised Emergent Autonomous Decision Making

2.1 Framework

The situation considered involves a robotic agent, a human supervisor of the agent, and a human with which the agent, but not the supervisor, should interact. The agent proposes actions that are accepted or rejected by the supervisor prior to executing them. The method proposed in this paper aims at enabling the agent to progressively and autonomously approximate the ideal behaviour as specified by the supervisor.

Our framework has five components: an agent and an environment interacting with each other, a supervisor, the algorithm controlling the agent and a context characterising the interaction between the agent and the environment. The agent has a defined set of available actions $\mathcal{A}$. The environment could be a human, a robot, or a computer for example and is characterised by a n-dimensional vector $\mathcal{S} \in \mathcal{R}^n$, which is time varying. The context $\mathcal{C} \in \mathcal{R}^m$ gives a set of parameters defining higher-level aspects, such as goals or the state, of the interaction, see figure 1. The supervisor and the agent have direct access to the context, but may ignore the real value of the state and see it only through observations.

The principal constraints are that the interaction has one or more high level goals, and some available actions can have a negative impact on these goals if executed in specific states. This should be avoided, so algorithms depending on randomness to explore the environment state space are inappropriate.

In order to simplify the system, we are making the following assumptions. Firstly, that the environment, while dynamic, is consistent: it follows a defined set of rules $\mathcal{E}$ which also describe how the context is updated. Secondly, the supervisor $\mathcal{T}$ is omniscient (complete knowledge of the environment), constant (does not adapt during the interaction), and coherent (will react with the same action if two sets of inputs are identical). Finally, the supervisor has some prior knowledge of the environment $\mathcal{K}$.

The algorithm has a model $\mathcal{M}$ of the supervisor and the environment and will update it through online learning following the learning method $\mathcal{L}$. $\mathcal{M}$ is iteratively updated based on supervisor feedback to approximate $\mathcal{T}$ and $\mathcal{E}$, in this way progressively approximating the action that the supervisor would have chosen, and what impact this would have on the environment. Equation 1 describes the update of each part of the framework from the step n to $n+1$.

$$\begin{aligned} M_n &: (C_{0\to n}, S_{0\to n}, A_{0\to n-1}, A'_{0\to n-1}) \longrightarrow A'_n \\ \mathcal{T} &: (C_{0\to n}, S_{0\to n}, A'_{0\to n}, A_{\to n-1}, \mathcal{K}) \longrightarrow A_n \\ \mathcal{E} &: (C_{0\to n}, S_{0\to n}, A_{0\to n}) \longrightarrow (S_{n+1}, C_{n+1}) \\ \mathcal{L} &: (C_{0\to n+1}, S_{0\to n+1}, A'_{0\to n}, A_{0\to n}) \longrightarrow M_{n+1} \end{aligned} \tag{1}$$

At the start of the interaction, the environment is in a state S_0 with the context C_0 and the algorithm has a model M_0. Applying M_0 to C_0 and S_0, the algorithm will select an action A'_0 and propose it to the supervisor. The supervisor can either accept this action or select a new one according to $\mathcal{T}$, and makes the agent execute the resulting action A_0. The environment will change to a new state S_1 and the context will be updated to C_1 according to $\mathcal{E}$. Based on S_1, S_0, C_1, C_0, A_0, and A'_0, the algorithm will update its model to M_1. The process can then be repeated based on the updated model.

2.2 Related Work

The approach we take here necessarily requires the application of machine learning, but we do not commit at this stage to a single algorithmic approach; the specific requirements for our application include online learning, deferring to an external supervisor, and being able to handle a dynamic environment.

A widely used method to transfer knowledge from a human to a robot is Learning from Demonstration (LfD), see [2] for a survey. In the case of policy learning, a teacher provides the learning algorithm with correct actions for the current state and repeats this state-action mapping for enough different states to give the algorithm a general policy. LfD is usually combined with supervised learning: trying directly to map outputs and inputs from a teacher, see [12] for a list of algorithms that can be used in supervised learning. The other important point is how the demonstrations are generated, a first approach is using batch learning: the teacher trains the algorithm during a training phase after which the robot is used in full autonomy [11]. Or there may be no explicit training phase; using online learning the demonstrations are given during the execution if required: the robot can request a demonstration for the uncertain states, e.g. when a confidence value about the action to perform is too low [6].

Another method is Reinforcement Learning: the algorithm tries to find a policy maximising the expected reward [3, 10]. However, this implies the presence of a reward function, which may not be trivial to describe in domains (such as social interaction) that do not lend themselves to characterisation. Consequently Abbeel and Ng proposed to use Inverse Reinforcement Learning by using an expert to generate the reward function [1], subsequently extended to use partially-observable MDPs [8], although expert-generated rewards also pose problems on the human side [17].

The goal of our proposed approach differs from these alternative existing methods. The intention is to provide a system that can take advantage of expert human knowledge to progressively improve its competencies without requiring manual intervention on every interaction cycle. This is achieved by only asking the human supervisor to intervene with corrective information when the proposed action of the robot agent is deemed inappropriate (e.g. dangerous) prior to actual execution. This allows the robot to learn from constrained exploration; a consequence of this is that the load on the supervisor should reduce over time as the robot learns. The supervisor nevertheless retains control of the robot, and as such we characterise the robot as having *supervised autonomy.* Contrary to the active learning approach used by, for example, Cakmak and Thomaz [5] the robot is not asking a question and requiring a supervisor response, it is proposing an action which may or may not be corrected by the supervisor.

3 Case Study: Application to Therapy

One potential application area is Robot Assisted Therapy (RAT) for children with Autism Spectrum Disorders (ASD). Children with ASD generally lack typical social skills, and RAT can help them to acquire these competencies, with a certain degree of success, e.g. [7, 14]. However, these experiments typically use the Wizard of Oz (WoZ) paradigm [13], which necessitates a heavy load on highly trained human operators.

We propose the use of supervised autonomy [15, 16], where the robot is primarily autonomous, but the therapeutic goals are set by a therapist who maintains oversight of the interaction. Having a supervised autonomous robot would reduce the workload on the therapist.Both the therapist and the robot would be present in the interaction, the robot interacting with the child and the therapist supervising the interaction and guiding the robot while it is learning its action selection policy.

The formalism described above (section 2.1) can be directly applied to this scenario. In this case, the context is the state of the task selected by the therapist to help the child develop certain social competencies, for example, a collaborative categorisation game [4] intended to allow the child to practice turn taking or emotion recognition. The state may be defined using multiple variables such as motivation, engagement, and performance exhibited by the child during the interaction, the time elapsed since the last child's action, and their last move (correct or incorrect). The robot could have a set of actions related to the game, such as proposing that the child categorises an image, or giving encouragement to the child.

In this scenario, the goal would be to allow the child to improve their performance on the categorisation task, and this would be done by selecting the appropriate difficulty level and finding a way to motivate the child to play the proposed game. We can expect the child to react to the robot action and that these reactions can be captured by the different variables that define the child's state (as provided by therapists for example). In principle, while precise determinations are likely to be problematic, we assume that some aspects of these variables can be estimated using a set of sensors (e.g. cameras and RGBD sensors to capture the child's gaze and position; the way the child interacts with the touch screen; etc). For the remainder of this paper, however, we assume that a direct estimation of internal child states are available to the system.

3.1 Proof of concept

A minimal simulation was constructed to illustrate the case study described above. The state $\mathcal{S}$ is defined using three variables: the child's performance, engagement and motivation in the interaction. The robot has the following set of actions $\mathcal{A}$: encouragement (give a motivating feedback to the child), waving (perform a gesture to catch the child's attention), and proposition (inviting the child to make a classification). In this minimal example, the environment $\mathcal{E}$ is the child model. A minimal model of the child was constructed that encompassed both processes that were dependent on the robot behaviour (e.g. responding to a request for action), and processes that were independent of the robot behaviour (e.g. a monotonic decrease of engagement and motivation over time independently of other events). The reaction of the model follows a rule-based system, but the amplitude of the response is randomly drawn from a normal distribution to represent the stochastic aspect of the child's reaction and the potential influence of non-defined variables in the state. A number of simplifications are necessary, such as the assumption of strict turn-taking and interactions in discrete time. The minimal child model is summarised in figure 2.

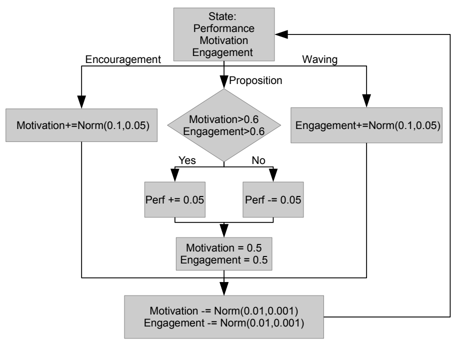

Figure 2: Model of the child used in the minimal simulation; random numbers are drawn from normal distributions.

Formally, the minimal simulation follows the framework established above (equation 1), with the simplification that a history of prior states, contexts, and actions is not used in the learning algorithm. This results in a setup where the system makes a suggestion of an action to take, which the supervisor can either accept or reject, in which case an alternative action is chosen (figure 3).

This allows the supervisor to take a more passive approach when the algorithm selects an acceptable action since they will only have to manually select a corrective action when this is needed. If the learning method is effective, the number of corrective actions should decrease over time, decreasing the workload on the therapist over the interaction.

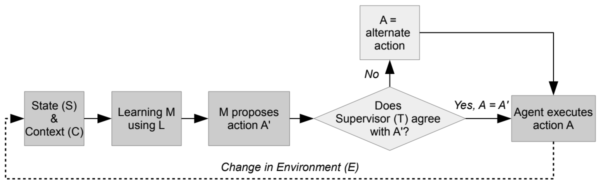

Figure 3: Description of agent's action selection process: the agent proposes actions that are validated by the supervisor prior to execution.

The learning model $\mathcal{M}$ is a MultiLayer Perceptron (MLP), with three input nodes for the input states, three output nodes for the three possible actions and nine nodes in the hidden layer. The model is trained using backpropagation (as $\mathcal{L}$), the true labels are given by the supervisor decision: output of 1 for the action selected by the supervisor (A) and -1 for the other ones. A Winner-Takes-All process is applied on the output of the MLP to select the action suggested by the robot (A').

Figure 4 shows a subset of a run from step 100 to 150. With this approach, there is no distinct learning and testing phases, but in the first part of the interaction (before step 100), the supervisor had to produce multiple corrective actions to train the network to express the desired output. The strategy used by the supervisor is the following: if the motivation is lower than 0.6 the supervisor enforces the action 'encouragement', else if the engagement is below 0.6 'waving' is enforced, and if both are above 0.6 then a proposition is made. The first graph presents the evolution of the state over time, and the second one the output of the MLP for each action. The vertical red lines represent an intervention from the supervisor, i.e. a case where the supervisor enforces a different action than the one suggested by the MLP. The action actually executed is represented by a cross with the same colour as the respective curves.

Figure 5 shows a comparison of the cumulative total of the different actions suggested and of the intervention as well as the child performance for three different models of a child and for a random action selection scheme. The difference in the child models in the three first graphs is the value of the thresholds required for a good classification action, high reactive child: 0.6 and 0.6, asymmetrically: 0.9 for encouragement and 0.6 for engagement, and low reactive: 0.9 and 0.9. Below these thresholds, a proposition would lead to a bad action decreasing the performance. It can be observed that the algorithm learns different strategies for each child and that there is more learning apparent at the start of the interaction than at the end (the rate of interventions is decreasing over time), indicating that the system is choosing the appropriate action at the appropriate time, and that the workload on the supervisor (necessity to provide these corrective actions) decrease over time. The last plot demonstrates a random action selection with a high reactive child. Contrary to the other cases, the child's performance decreases over time, and the number of interventions increases. Here, a *bad* action only decreases the performance, but in reality it may result in the termination of the interaction, which must be avoided.

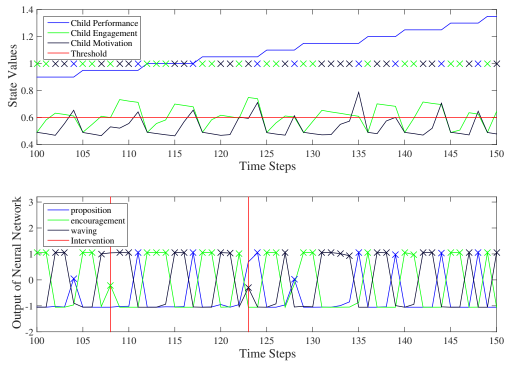

Figure 4: Subset of an interaction from step 100 to 150.

4 Discussion

While demonstrating promise, there are a number of limitations to the framework as presented. The assumptions described in section 2.1 are typically violated when working with humans. Firstly the children are all different, and a method learned for one child may often not be suited when working with another. Furthermore, the same child may have non-consistent behaviour between the sessions and even within a single session. There is no perfect solution to solve this problem, but we can expect that with enough training sessions and a more complex learning algorithm, the system would be able to capture patterns and react to the different behaviours appropriately. Since it is ex-

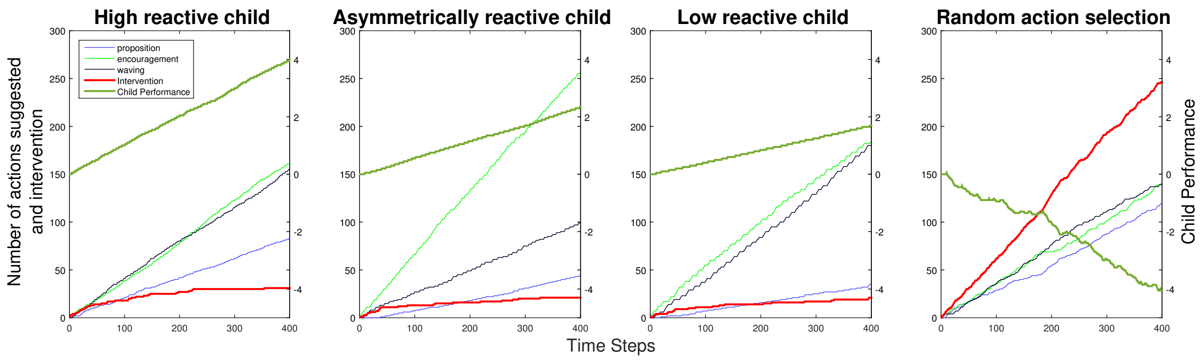

Figure 5: Comparison of the cumulative total of the different actions suggested, the supervisor interventions required, and child performance for three different models of a child (highly responsive; asymmetrically increased responsiveness to engagement than motivation; low responsiveness), and a random action selection scheme.

pected that in a real application of such an approach a therapist who knows the child will always be present, we propose that for a new child the algorithm will use a generic strategy based on previous interactions with other children, with subsequent fine-tuning under supervision.

Another assumption that is likely to be violated is that of a perfect supervisor. As explained in [6] humans are not always consistent nor omniscient, but methods presented in the literature can be used to cope with these inconsistencies if enough data is gathered. Further mitigating solutions could be employed, such as the robot warning the therapist if it is about to select an action which had negative consequences in a previous interaction (even if for a different child). Furthermore, it may not be possible to measure the true internal states of the child in the real world, with imperfect estimations of these states being more likely accessible. In this case, inspiration from [9] can be used to mixed the POMDP framework with the help of an exterior oracle. Another problem which will have to be addressed in the future is the difference in inputs between the robot and the therapist: the therapist will have access to language, more subtle visual features and their prior experience, whereas the robot may have direct and precise access to some aspects of the child's overt behaviour (such as timings of touchscreen interaction).

In the currently implemented case study, we assume that the supervisor responds to the action proposed by the robot within some predetermined fixed time, whether this response is accept or reject (figure 3). This, in principle, allows the supervisor to only actively respond if a proposed action is clearly inappropriate. In further developments, we will incorporate a measure of certainty (given prior experience) into the time allowed the supervisor to respond to the proposed action: for example increasing the time available if the confidence in the proposed action is low. This modulation of the load on the supervisor's attention according to confidence should result in the supervisor being able to increasingly pay attention to the child directly, rather than to the robot system, as training progresses.

5 Conclusion

We have presented a general framework to progressively increase the competence of an autonomous action selection mechanism that takes advantage of the expert knowledge of a human supervisor to prevent inappropriate behaviour during training. This method is particularly applicable to application contexts such as robot-assisted therapy, and our case study has provided preliminary support for the utility of the approach. While the simulation necessarily only provided a minimal setup, and thus omitted many of the complexities present in a real-world setup, we have nevertheless shown how the proposed method resulted in the learning of distinct action selection strategies given differing interaction contexts, although further refinement is required for real-world application. Indeed, given real-world supervisor knowledge limitations, we suggest it will furthermore be possible for a suitably trained action selection mechanism of this type to aid the supervisor in complex and highly dynamic scenarios.

Acknowledgements

This work is supported by the EU FP7 DREAM project (grant 611391).

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 1–8, 2004.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[3] A. G. Barto. *Reinforcement learning: An introduction.* MIT press, 1998.

[4] P. Baxter, R. Wood, and T. Belpaeme. A touchscreen-based sandtrayto facilitate, mediate and contextualise human-robot social interaction. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pages 105–106. IEEE, 2012.

[5] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[6] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.

[7] K. Dautenhahn. Robots as social actors: Aurora and the case of autism. In *Proc. CT99, The Third International Cognitive Technology Conference, August, San Francisco*, volume 359, page 374, 1999.

[8] F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using bayes risk for active learning in pomdps. In *Proceedings of the 25th international conference on Machine learning*, pages 256–263. ACM, 2008.

[9] F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using bayes risk for active learning in pomdps. In *Proceedings of the 25th international conference on Machine learning*, pages 256–263. ACM, 2008.

[10] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, pages 237–285, 1996.

[11] W. B. Knox, S. Spaulding, and C. Breazeal. Learning social interaction from the wizard: A proposal. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[12] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[13] L. Riek. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, 1(1):119–136, Aug. 2012.

[14] B. Robins, K. Dautenhahn, R. T. Boekhorst, and A. Billard. Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills? *Universal Access in the Information Society*, 4(2):105–120, July 2005.

[15] E. Senft, P. Baxter, J. Kennedy, and T. Belpaeme. When is it better to give up?: Towards autonomous action selection for robot assisted asd therapy. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, HRI'15 Extended Abstracts, pages 197–198, New York, NY, USA, 2015. ACM.

[16] S. Thill, C. A. Pop, T. Belpaeme, T. Ziemke, and B. Vanderborght. Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, 3(4):209–217, Apr. 2013.

[17] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008.

SPARC: Supervised Progressively Autonomous Robot Competencies

Emmanuel Senft, Paul Baxter, James Kennedy, and Tony Belpaeme

Centre for Robotics and Neural Systems, Plymouth University, United Kingdom
{emmanuel.senft, paul.baxter, james.kennedy,
tony.belpaeme}@plymouth.ac.uk

Abstract. The Wizard-of-Oz robot control methodology is widely used and typically places a high burden of effort and attention on the human supervisor to ensure appropriate robot behaviour, which may distract from other aspects of the task engaged in. We propose that this load can be reduced by enabling the robot to learn online from the guidance of the supervisor to become progressively more autonomous: Supervised Progressively Autonomous Robot Competencies (SPARC). Applying this concept to the domain of Robot Assisted Therapy (RAT) for children with Autistic Spectrum Disorder, a novel methodology is employed to assess the effect of a learning robot on the workload of the human supervisor. A user study shows that controlling a learning robot enables supervisors to achieve similar task performance as with a non-learning robot, but with both fewer interventions and a reduced perception of workload. These results demonstrate the utility of the SPARC concept and its potential effectiveness to reduce load on human WoZ supervisors.

1 Introduction

Over the last two decades, an increasing amount of research has been conducted to explore Robot Assisted Therapy (RAT). Using robots in therapies for children with Autism Spectrum Disorder (ASD) has revealed promising results [5, 10, 11]. The Wizard-of-Oz (WoZ) paradigm is typically used for this application, and others, where the robots are not autonomous but tele-operated. Many motivating factors for moving away from WoZ in RAT have been put forward [8, 13]. In particular, autonomous behaviour facilitates repetition of the robot behaviour and decreases the workload on therapists, freeing them to pay attention to other aspects of the interaction. It is the intention of our research to facilitate this shift to robot autonomy.

As the optimal robot behaviour is unlikely to be known in advance (be it in a therapeutic or indeed other domain), and with adaptability during and between the different interactions being generally desirable, it is necessary to provide the robot with learning capabilities. In the context of RAT, by using the knowledge of a therapist, the learning can be guided so that it is faster and safer, especially as the robot cannot use random exploration to acquire knowledge about its environment when interacting with children with ASD in case of negative therapeutic and/or clinical outcomes. We propose an approach taking inspiration from the Learning from Demonstration and online learning literature, and call it SPARC: Supervised Progressively Autonomous Robot Competencies. In SPARC, a therapist guides the robot in the early stages of the interaction, and progressively, the robot learns an action policy adapted to the particular therapeutic session [12]. Assuming the effective learning of the robot in this context, the therapist can allow the robot to behave increasingly autonomously, whilst maintaining oversight. Although not reducing the attentional requirements, this would reduce the physical interventions to direct the robot behaviour required by the therapist. Thus, by proposing and executing good actions, SPARC can reduce the therapists' workload.

A RAT scenario typically involves three parties: the patient, a robot, and the human therapist. In this context, the therapist does not interact with the patient directly, but rather through the robot. The therapist could therefore be described as playing the role of a robot supervisor. The focus of this paper is not on a new learning algorithm, but rather on the interaction between the robot and the therapist (supervisor), and the role that robot autonomy can play in this relationship. Specifically, as an initial validation of the principle, we seek to assess whether the SPARC concept can feasibly result in a reduction in workload for the supervisor, even given different strategies used by different individuals. A user study employing a novel methodology is conducted (section 3), demonstrating that progressive robot autonomy does indeed result in lower supervisor workload (section 4). This outcome provides support for the proposed approach and motivates further development efforts in the domain of RAT.

2 Related work

A number of research groups have studied the use of robot in therapy for children with ASD, which allowed children to express previously unseen social behaviour for example [9, 10]. Two primary methods have been used for these investigations: using an autonomous robot following preprogrammed rules [6, 14], or using the WoZ paradigm, allowing more flexibility in the robot's reaction. As noted in [8, 13], using WoZ allows testing and prototyping of interaction scenarios, but researchers should consider moving away from it to achieve more scalability, more repeatability, and to allow the use of robots without increasing the workload on therapists. Complex behaviour is required for a therapeutic robot, thereby making learning a desirable feature for future, more autonomous, RAT. As therapists possess the knowledge required to make appropriate decisions in different contexts, Learning from Demonstration [1] provides a useful starting point. Recently, Knox *et al.* proposed the Learning from Wizard paradigm in [7]. The robot is first controlled by a human operator as in a WoZ scenario, and after a number of interactions, batch learning is applied on the previous interaction data to obtain autonomous behaviour.

A fixed action policy of this type is however not desirable for RAT as children may not be consistent between interactions, and thus online learning is required

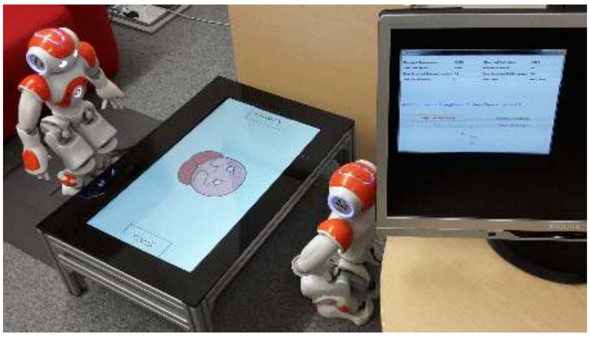

Fig. 1. Setup used for the user study from the perspective of the human supervisor. The *child-robot* (left) stands across the touchscreen (centre-left) from the *wizarded-robot* (centre-right). The supervisor can oversee the actions of the *wizarded-robot* through the GUI and intervene if necessary (right).

to provide the robot with the adaptability necessary to update its action policy depending on the current circumstances. Several experimenters in HRI have studied active learning: a robot actively questions a human teacher in order to request data points or demonstration for an uncertain scenario. A study exploring the type of questions that a robot could ask and the human reactions can be seen in [3], and Chernova and Veloso propose a progressive learning algorithm where a robot can estimate the confidence in its action decision in a fixed environment [4]: if the confidence is too low, a demonstration from a human teacher is required to complete the task.

However, an important element missing from the current literature is online learning for interaction. The robot needs to be able to progressively create an action policy, and update it later if necessary, to reach a more complex interaction behaviour. This paper explores how supervised progressive learning can be used in an interaction scenario and introduces a novel methodology to test this technique.

3 Assessing the effect of a progressively autonomous robot on supervisor workload

The focus of the present study is to assess whether the application of the SPARC concept to RAT results in a decrease in workload for the human supervisor. Two types of robot controller are employed to determine the presence and magnitude of this effect: a robot that learns from the actions of the supervisor to progressively improve its behaviour (*learning* controller), and a robot that only generates random actions (*non-learning* controller).

The methodology used in this paper is based on a real scenario for RAT for children with ASD based on the Applied Behaviour Analysis therapy framework. The aim of the therapy is to help the child to develop/practice their social skills: the task we focus on here is emotion recognition. This scenario involves a child playing a categorisation game with a robot on a mediating touchscreen device [2]. Images of faces or drawings are shown to the child, and she has to categorise them by moving the image to one side or the other depending on whether the picture shown denotes happiness or sadness (e.g. fig. 1). The human supervisor is physically present and guides the robot using the Wizard of Oz paradigm, but does not interact with the child directly.

In our proposed system, the basic interaction structure following the SPARC concept is as follows: the robot suggests an action to the supervisor, the supervisor agrees or disagrees with this suggestion (providing an alternative if disagreeing), the robot executes the action, and then both robot and supervisor observe the outcome. Over time, it is possible for the robot to learn an appropriate strategy based on observations of the child and oversight from the supervisor, with the supervisor still maintaining overall control if necessary.

Given the focus on human supervisor workload, it is necessary to provide a consistent experimental environment across both conditions in which the task, setup, and interaction partner is kept constant. A minimal model of child behaviour is therefore used to stand in for a real child. A second robot is employed in the interaction to embody this child model: we term this the *child-robot*. The robot being directly guided by the human supervisor is termed the *wizarded-robot* (fig. 1).

3.1 Child model

The purpose of the child model is not to realistically model a child (with or without autism), but to provide a means of expressing some of the behaviours we observed in our interactions with children in a repeatable manner. The child-robot possesses an internal model encompassing an *engagement* level and a *motivation* level, together forming the *state* of the child. The engagement represents how often the child-robot will make categorisation moves and the motivation gives the probability of success of the categorisation moves. Bound to the range $[-1, 1]$, these states are influenced by the behaviour of the wizarded-robot, and will asymptotically decay to zero without any actions from the wizarded-robot. These two states are not directly accessed by either the supervisor or the wizarded-robot, but can be observed through behaviour expressed by the child-robot: low engagement will make the robot look away from the touchscreen, and the speed of the categorisation moves is related to the motivation (to which gaussian noise was added). There is thus incomplete/unreliable information available to both the wizarded-robot and the supervisor, making the task non-trivial.

The influence of the wizarded-robot behaviour on the levels of engagement and motivation are described below (section 3.2). In addition to this, if a state is already high and an action from the wizarded-robot further increases it, then there is a chance that this level will sharply decrease, as an analogue of child-robot *frustration*. When this happens, the child-robot will indicate this frustration verbally (uttering one of eight predefined strings). The reason this mechanism is required is that it prevents a straightforward engagement and motivation maximisation strategy, thus better approximating the real situation, and requiring a more complex strategy to be employed by the supervisor.

3.2 Wizarded-robot control

The wizarded-robot is controlled through a Graphical User Interface (GUI) and has access to multiple variables characterising the state of the interaction. The wizarded-robot has a set of four actions, which each have a button in the GUI:

- Prompt an Action: Encourage the child-robot to do an action.
- Positive Feedback: Congratulate the child-robot on making a good classification.
- Negative Feedback: Supportive feedback for an incorrect classification.
- Wait: Do nothing for this action opportunity, wait for the next one.

The impact of the action on the child-robot depends on the internal state and the type of the last child-robot move: good, bad, or done (meaning that feedback has already been given for the last move and supplementary feedback is not necessary). A prompt always increases the engagement, a wait has no effect on the child-robot's state, and the impact of positive and negative feedback depends on the previous child-robot move. Congruous feedback (positive feedback for correct moves; negative feedback for incorrect moves) results in an increase in motivation, but incongruous feedback can decrease both the motivation and the engagement of the child-robot. The supervisor therefore has to use congruous feedback and prompts, whilst being careful not to use them too often, to prevent the child-robot becoming frustrated. A 'good' strategy would keep the engagement and motivation high, leading to an increase in performance of the child-robot in the categorisation task.

Through the GUI, the supervisor has access to *observed states* (noisy estimations of the child-robot state), and information about the interaction history: number of moves, child-robot performance, time since last child-robot and wizarded-robot actions, type of the last child-robot move, and elapsed time. However the supervisor can not control the wizarded-robot directly, actions can only be executed only at specific times triggered by the wizarded-robot. Two seconds after each child-robot action, or if nothing happens in the interaction for five seconds, the wizarded-robot proposes an action to the supervisor by displaying the action's name and a countdown before execution. Only after this proposition has been done can the supervisor provide feedback to the wizarded-robot. If the supervisor does nothing in the following three seconds, the action proposed by the wizarded-robot is executed. This mechanism allows the supervisor to passively accept a suggestion made by the wizarded-robot or actively make an *intervention* by selecting a different action and forcing the wizarded-robot to execute it.

3.3 Learning algorithm

The two robot controllers used for the study were a learning controller and a non-learning random action selection controller. The learning algorithm used was a Multi-Layer Perceptron, trained with back propagation (five input, six hidden and four output nodes): after each new decision from the supervisor, the network was fully retrained with all the previous state-action pairs and the new one.

3.4 Participants

In WoZ scenarios, the wizard is typically a technically competent person with previous experience controlling robots. As such, to maintain consistency with the target user group, the participants for this study (assuming the role of the supervisor) are taken from a robotics research group. Ten participants were used (7M/3F, age M=29.3, 21 to 44, SD=4.8 years).

3.5 Hypotheses

To evaluate the validity of our method and the influence of such an approach, four hypotheses were devised:

H1 A 'good' supervisor (i.e. keeping the motivation and engagement of the child-robot high) will lead to a better child-robot performance.
H2 When interacting with a new system, humans will progressively build a personal strategy that they will use in subsequent interactions.
H3 Reducing the number of interventions required from a supervisor will reduce their perceived workload.
H4 Using a learning wizarded-robot allows the supervisor to achieve similar performance with fewer interventions when compared to the same scenario with a non-learning wizarded-robot.

3.6 Interaction Protocol

Each participant experienced both robot controllers, with the order changed between participants to control for any ordering effects. In *Condition LN* the participants first interact with the learning wizarded-robot, and then with the non-learning one; in *Condition NL* the participants first interact with the non-learning wizarded-robot, and then the learning robot. Participants were randomly assigned to one of the two conditions.

The interactions took place on a university campus in a dedicated experiment room. Two Aldebaran Nao robots were used; one robot had a label indicating that it was the *Child-Robot*. The robots face each other with a touchscreen between them, and participants assuming the role of the supervisor sit at a desk to the side of the wizarded-robot, with a screen and a mouse to interact with the wizarded-robot (fig. 1). The participants were able to see the screen and the child-robot.

A document explaining the interaction scenario was provided to participants. After the information had been read, a 30s video presenting the GUI in use was shown to familiarise them with it, without biasing them towards any particular intervention strategy. The participant then clicked a button to start the first interaction which lasted for 10 minutes. The experimenter was sat in the room outside of the participants' field of view. After the end of the first interaction, a post-interaction questionnaire was administered. The same protocol was applied in the second part of the experiment with another post-interaction questionnaire following. Finally, a questionnaire asking the participants to explicitly compare the two conditions was administered.

4 Results

4.1 Interaction data

The state of the child and the interaction values were logged at each step of the interaction (at 5Hz). All of the human actions were recorded: acceptance of the wizarded-robot's suggestion, selection of another action (intervention), and the states of the child-robot (motivation, engagement and performance) at this step. From this the intervention ratio was derived: the number of times a user chose a different action to the one proposed by the wizarded-robot, divided by the total number of executed actions. On average, after a first exploration phase, where the participant discovers the system, the learning robot robot has an intervention ratio lower than the non learning one (fig. 2, left)

The performance indicates the number of good categorisations executed by the child-robot minus the number of bad categorisations. A strong positive correlation (Pearson's r=0.79) was found between the average child-robot motivation and engagement and its performance.

In both conditions, the average performance in the second interaction (M_{LN-2} =38, 95% CI [36.2, 39.8], M_{NL-2}=34.8, 95% CI [30.8, 38.8]) was higher than in the first one (M_{LN-1}=29.4, 95% CI [25.3, 33.5], M_{NL-1}=24.3, 95% CI [19.4, 29.4]; Fig. 2 *left*). The 95% *Confidence Interval of the Difference of the Mean* (CIDM) for the L-NL condition is [4.1, 13.1] and for the NL-L condition is [4.0, 16.8]. However, the performance is similar when only the interaction order (first or second) is considered. The participants performed slightly better in the LN condition, but the CIDM includes zero in both cases (95% CIDM_1 [-1.5, 11.5], 95% CIDM_2 [-1.2, 7.6]). In the condition L-NL, the intervention ratio increased between the learning and non learning condition (M_{LN-1}=0.31, 95% CI [0.20, 0.42] to M_{LN-2}=0.68, 95% CI [0.66, 0.70], CIDM_{LN}=[0.26, 0.48]). But in the NL condition, the intervention ratio is almost identical between the two interactions but slightly lower for the learning case (M_{NL-1}=0.50, 95% CI [0.44, 0.57] to M_{NL-2}=0.46, 95% CI [0.40, 0.51], CIDM_{NL} [-0.03, 0.13]). This shows that when the wizarded-robot learned, a similar performance is attained as without learning, but the number of interventions required to achieve this is lower.

4.2 Questionnaire data

The post-interaction questionnaires evaluated the participant's perception of the child-robot's learning and performance, the quality of suggestions made by the

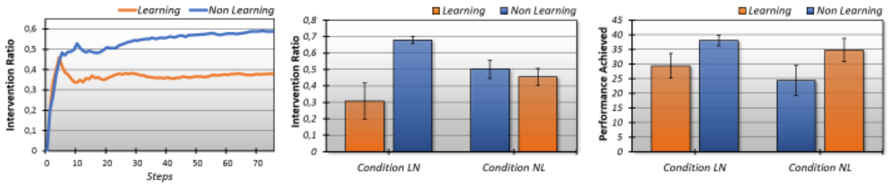

Fig. 2. (Left) evolution of intervention ratio over time for the learning and non learning cases. Intervention ratio (centre) and final performance (right) for the two conditions and the two interactions (*errors bars show 95% CI*). In condition LN participants started wizarding a robot which learns their interaction style, followed by a non-learning robot; in condition NL participants started with a non-learning robot, followed by a learning robot. Results show that a learning robot reduces the workload of the wizard, but performs equally well as a non-learning robot that needs wizarding at all times.

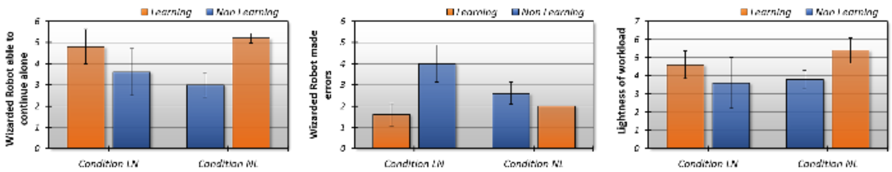

Fig. 3. Questionnaire responses (*mean and 95% CI*): increased confidence in the learning wizarded-robot over the non-learning version is apparent, as is a lower perceived workload.

wizarded-robot, and the experienced workload. All responses used seven point Likert scales.

Across the four possible interactions, the rating of the child-robot's learning was similar (M=5.25, 95% CI [4.8, 5.7]). The same effect was observed for the evaluation of the child performance (M=4.75, 95% CI [4.3, 5.2]). As the child-robot was using the same interaction model in all four conditions, this result is expected.

Participants report the wizarded-robot as more suited to operate unsupervised in the learning than in the non learning condition (M_{LN-1}=4.8, M_{LN-2}=3.6, M_{NL-1}=3, M_{NL-2}=5.2 ; CIDM for LN condition [-0.2, 2.6], CIDM for the NL condition [1.6, 2.8]).

Similarly, a trend was found showing that learning wizarded-robot is perceived as making fewer errors than the non-learning robot (M_{LN-1}=1.6, M_{LN-2}=4.0, M_{NL-1}=2.6, M_{NL-2}=2 ; CIDM for LN condition [1.3, 3.4], CIDM for the NL condition [0.1, 1.1]).

The participants tended to rate the workload as lighter when interacting with the learning robot, and this effect is much more prominent when the participants interacted with the non-learning robot first (M_{LN-1}=4.6, M_{LN-2}=3.6, M_{NL-1}=3.8, M_{NL-2}=5.4 ; CIDM for LN condition [-0.6, 2.6], CIDM for the NL condition [0.7, 2.5]).

5 Discussion

Strong support for H1 (a good supervisor leads to a better child performance) was found, a correlation between the average states (engagement and motivation) and the final performance for all of the 10 participants was observed (r=0.79). We could expect a similar effect when working with real children, but measuring these values would be a challenge.

The results also provide support for H2 (supervisors create personal strategies): all the participants performed better in the second interaction than in the first one. This suggests that participants developed a strategy when interacting with the system in the first interaction, and were able to use it to increase their performance in the second interaction. Looking in more detail at the interaction logs, it is possible to see that different people used different strategies.

H3 (reducing the number of interventions will reduce the perceived workload) is partially supported: the results show a trend for participants to rate the workload as lighter when interacting with the learning robot, and another trend between using a learning robot and the intervention ratio. However, when considering the difference of workload rating and intervention ratios between the two interactions, a positive correlation is only found for the LN condition, which could be accounted for by the initial steep learning curve for the study participants. Nevertheless, regardless of the order of the interactions, the learning robot consistently received higher ratings for lightness of workload (fig. 3).

Finally, H4 (using learning keeps similar performance, but decreases interventions) is supported: interacting with a learning robot results in a similar performance than interacting with a non-learning robot, whilst requiring fewer active interventions from the supervisor. This has real world utility, it frees some time for the supervisor, to allow her to focus on other aspects of the intervention, e.g. analysing the child's behaviour rather than focusing on the robot control.

It should be noted that the actual learning algorithm used in this study is only of incidental importance, and that certain features of the supervisor's strategies may be better approximated with alternative methods – of importance for the present work is the presence of learning at all. Future work will assess what the most appropriate machine learning approach is given the observed features of supervisor strategy from this study.

In conclusion, this paper proposed the SPARC concept (Supervised Progressively Autonomous Robot Competencies). Based on a suggestion/intervention system, this approach allows online learning for interactive scenarios, thus increasing autonomy and reducing the demands on the supervisor. Results showed that interacting with a learning robot allowed participants to achieve a similar performance as interacting with a non-learning robot, but requiring fewer interventions to attain this result. This suggests that while there is always adaptation in the interaction (leading to similar child-robot performance given the two wizarded-robot controllers), the presence of learning shifts this burden of adaptivity onto the wizarded-robot rather than on the human. This indicates that a learning robot could allow the therapist to focus more on the child than on the robot, with improved therapeutic outcomes as potential result.

Acknowledgements

This research was funded by the EU FP7 DREAM project (grant no. 611391).

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and autonomous systems 57(5), 469–483 (2009)
2. Baxter, P., Wood, R., Belpaeme, T.: A touchscreen-based sandtray to facilitate, mediate and contextualise human-robot social interaction. In: Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on. pp. 105–106. IEEE (2012)
3. Cakmak, M., Thomaz, A.L.: Designing robot learners that ask good questions. In: Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction. pp. 17–24. ACM (2012)
4. Chernova, S., Veloso, M.: Interactive policy learning through confidence-based autonomy. Journal of Artificial Intelligence Research 34(1), 1 (2009)
5. Dautenhahn, K.: Robots as social actors: Aurora and the case of autism. In: Proc. CT99, The Third International Cognitive Technology Conference, August, San Francisco. vol. 359, p. 374 (1999)
6. Feil-Seifer, D., Mataric, M.: B3IA: A control architecture for autonomous robot-assisted behavior intervention for children with Autism Spectrum Disorders. RO-MAN 2008. (2008), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4600687
7. Knox, W.B., Spaulding, S., Breazeal, C.: Learning social interaction from the wizard: A proposal. In: Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
8. Riek, L.: Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. Journal of Human-Robot Interaction 1(1), 119–136 (Aug 2012)
9. Robins, B., Dautenhahn, K., Boekhorst, R.T., Billard, A.: Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills? Universal Access in the Information Society 4(2), 105–120 (Jul 2005)
10. Robins, B., Dautenhahn, K., Dickerson, P.: From Isolation to Communication: A Case Study Evaluation of Robot Assisted Play for Children with Autism with a Minimally Expressive Humanoid Robot. Conferences on Advances in Computer-Human Interactions pp. 205–211 (Feb 2009)
11. Scassellati, B., Admoni, H., Mataric, M.: Robots for use in autism research. Annual review of biomedical engineering 14, 275–294 (2012)
12. Senft, E., Baxter, P., Belpaeme, T.: Human-guided learning of social action selection for robot-assisted therapy (2015), in press for 4th Workshop on Machine Learning for Interactive Systems
13. Thill, S., Pop, C.A., Belpaeme, T., Ziemke, T., Vanderborght, B.: Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. Paladyn 3(4), 209–217 (2012)
14. Wainer, J., Dautenhahn, K., Robins, B., Amirabdollahian, F.: Collaborating with kaspar: Using an autonomous humanoid robot to foster cooperative dyadic play among children with autism. In: Humanoid Robots (Humanoids), 2010. pp. 631–638. IEEE (2010)

Providing a Robot with Learning Abilities Improves its Perception by Users

Emmanuel Senft, Paul Baxter, James Kennedy, Séverin Lemaignan and Tony Belpaeme
Centre for Robotics and Neural Systems
Plymouth University, U.K.
{emmanuel.senft, paul.baxter, james.kennedy, severin.lemaignan, tony.belpaeme}@plymouth.ac.uk

***Abstract*—Subjective appreciation and performance evaluation of a robot by users are two important dimensions for Human-Robot Interaction, especially as increasing numbers of people become involved with robots. As roboticists we have to carefully design robots to make the interaction as smooth and enjoyable as possible for the users, while maintaining good performance in the task assigned to the robot. In this paper, we examine the impact of providing a robot with learning capabilities on how users report the quality of the interaction in relation to objective performance. We show that humans tend to prefer interacting with a learning robot and will rate its capabilities higher even if the actual performance in the task was lower. We suggest that adding learning to a robot could reduce the apparent load felt by a user for a new task and improve the user's evaluation of the system, thus facilitating the integration of such robots into existing work flows.**

I. Introduction

This paper presents a study exploring the impact of providing a robot with learning capabilities on the interaction preferences and robot performance evaluations by users.

Two main approaches are reported in the literature to study human preferences about robots. The first one involves the administration of surveys where participants are asked robot-related questions with or without priming. For example, in [1] 240 subjects are asked questions about tasks that could be replaced by robots and about general attitudes toward robots, without trying to influence the participants *a priori*. Priming can also be a useful means of educating the participants before administering a questionnaire, allowing them to imagine a more constrained and plausible scenario than they otherwise would. This approach has been followed by Coeckelbergh et al. [2], who surveyed the attitudes of participants toward Robot Assisted Therapy (RAT) for children with autism spectrum disorder. The participants answered more positively, in contrast to previous studies conducted without priming, when they were first exposed to a one minute video presenting the state of the art of robotics in RAT.

The second main approach is administering a questionnaire to participants after an actual interaction with a robot. Using this method, the responses are grounded in the context of their interaction: this can diminish the generalisability of the results, but makes them more reliable. This method has been applied to explore how elderly people react to a robot with learning abilities [3].

This paper follows the real robot interaction approach, and presents additional results gathered in the experiment presented in [4]. In this study, participants interacted with a robot both with and without learning capabilities, and this manuscript reports their interaction preference and their relative performance evaluation of the two robots.

II. Methodology

The study (and therefore the methodology) is the same as in [4] where we introduced Supervised Progressively Autonomous Robot Competencies (SPARC), a means for the robot to learn from the interaction to improve its capabilities. This previous paper also reported the impact of SPARC on the performance and workload of a robot's human supervisor in a scenario inspired by RAT for children with autism spectrum disorder. In classical RAT, the robot is interacting with the child and is often controlled using the Wizard of Oz (WoZ) paradigm, i.e. fully tele-operated. This often implies a high workload on the therapist, which could be reduced by providing the robot with a supervised autonomy. As this study focuses on the interaction between the *wizarded-robot* and its supervisor, we replace the child with a robot interacting in his place (the *child-robot*) to produce consistent experimental conditions (fig. 1).

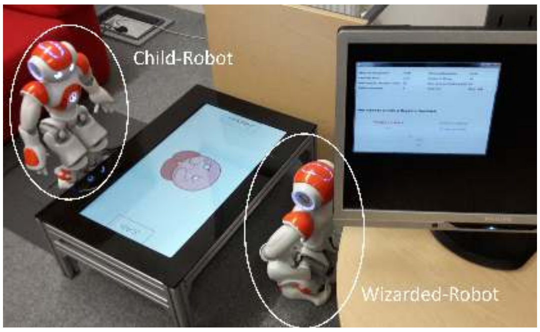

Fig. 1. Installation used for the study. The *child-robot* stands on the left, performing the task on the touchsceen, and facing the *wizarded-robot* on the centre-right. The human supervisor can control the action about to be executed by the *wizarded-robot* using the GUI on the right.

The child-robot is interacting with a touchscreen, and performs a categorisation task where it has to classify images of face as either happy or sad, with the aim of improving its performance. The wizarded-robot can execute actions (e.g. giving positive or negative feedback, waiting, or prompting the child to act), aiming to help the child-robot in its task.

The participants have to control the wizarded-robot to make it execute the correct actions to help the child-robot. This is however context dependent: actions can either improve or worsen the performance of the child-robot based on its current state. A Graphical User Interface (GUI) allows the users to control the wizarded-robot in a WoZ inspired scenario involving supervised autonomy. At specific times, the wizarded-robot makes suggestions to the supervisor who can either not react and let the action execute, or use a button to force the wizarded-robot to execute another action. A habituation phase allows the participants to become familiar with the interface and action set. If the suggestion of the robot is correct, the supervisor does not need to act to have this action executed.

The participants interacted with two systems. In the first system, the actions proposed by the wizarded-robot are random, so we expect the user to correct them most of the time. This system simulates a classical WoZ setup, which we denote the *non-learning robot*. The second system uses SPARC and includes a learning algorithm based on a Multi-Layer Perceptron using noisy observation of states as inputs and a winner-take-all on the actions as output. This system is referred to as the *learning-robot*. It is important to note that in both systems, these terms relate to the capabilities of the wizarded-robot, not the child-robot (which had constant behaviour in both systems).

The study involved ten participants (7M/3F, age M=29.3, 21 to 44, SD=4.8 years) taken from a robotic research group, as typical WoZ users are technical. Each participant interacted for 10 minutes with both systems, with the order counterbalanced. In the LN condition, participants interacted first with the learning robot then with the non-learning one, and the order is reversed in the NL condition. This paper presents and analyses the responses from the participants to the questions:

– Which *wizarded-robot* was better able to perform the task?
– Which *wizarded-robot* did you prefer supervising?

III. Results and Discussion

Overall, the participants preferred supervising the learning-robot (6 out of 10) and found it better able to perform the task (8 out of 10). Despite the limitations of the small sample size, these results suggest that providing a robot with learning capabilities can improve its perception by users and also make the users prefer supervising it. These results are consistent with previous results [4], which showed that providing a robot with learning capabilities can decrease the number of interventions required to achieve a similar performance compared to a robot without learning. The reduction in the number of interventions needed might explain the results observed here.

Breaking the results down by ordering condition (LN vs. NL) provides a more detailed perspective (fig. 2). From these separated results, the learning capability is not the only effect influencing the preferences of, and the evaluation by, the participants: the order of interaction also plays an important role. On average, the second robot is the preferred one to supervise (7 of 10) and rated as better able to perform the task (7 of 10). This ordering effect was probably due to the

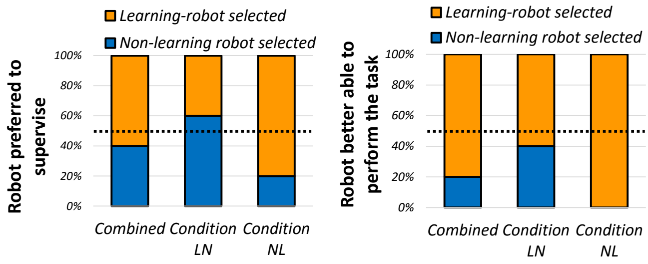

Fig. 2. Results for supervisory preference and rating of ‘preferred to supervise’ and ‘better to perform the task’ for the two conditions. The vertical bars represent the number of times that the learning robot was selected and the horizontal dotted line denotes chance (i.e. 50%).

complexity of the system that the participants interacted with. The participants had to get used to a GUI displaying a large volume of information, and to the time constraints.

Additionally, in 4 of the 5 cases when participants interacted with the learning robot first, they achieved a better performance during the second interaction than during the first one. Three of these participants also rated the learning robot as better able to perform the task even when it had a lower performance. This could indicate that participants can distinguish between the robot’s abilities and the performance achieved (depending also on their abilities). It could also be a reflection of the natural propensity of humans to adapt and learn through interaction. Viewed in this way, the results could be interpreted as showing that interaction with the learning robot first better equips the human to interact with the non-learning robot than vice-versa, leading to higher performance, and hence preference ratings, for the non-learning robot in the LN condition. While another potential benefit of learning robots, this interpretation will require further empirical investigation.

In this paper we presented results showing a trend towards the addition of learning capabilities to a robot helping users to cope with a new or complex task, and improving the rating of their performance by their supervisor. This is an important point for design, especially when there is a heavy workload on users such as in RAT when therapists have to use WoZ to continuously control the robot.

IV. Acknowledgements

This work is funded by the EU FP7 project DREAM (grant 611391).

References

[1] C. Ray *et al.*, “What do people expect from robots?” in *Int. Conf. on Intelligent Robots and Systems*. IEEE/RSJ, 2008.
[2] M. Coeckelbergh *et al.*, “A Survey of Expectations About the Role of Robots in Robot-Assisted Therapy for Children with ASD: Ethical Acceptability, Trust, Sociability, Appearance, and Attachment,” *Science and Engineering Ethics*, pp. 1–19, 2015.
[3] J. Hoefinghoff *et al.*, “”Yes Dear, that Belongs into the Shelf!”-Exploratory Studies with Elderly People Who Learn to Train an Adaptive Robot Companion,” in *Proc. of the 7th Int. Conf. on Social Robotics*, 2015.
[4] E. Senft *et al.*, “SPARC: Supervised Progressively Autonomous Robot Competencies,” in *Proc. of the 7th Int. Conf. on Social Robotics*, 2015.

Cognitive Architectures for Social Human-Robot Interaction

Paul Baxter, Séverin Lemaignan
Centre for Robotics and Neural Systems
Plymouth University, Plymouth, U.K.
{paul.baxter, severin.lemaignan}@plymouth.ac.uk

J. Gregory Trafton
Naval Research Laboratory
Washington DC, USA
greg.trafton@nrl.navy.mil

***Abstract*—Social HRI requires robots able to use appropriate, adaptive and contingent behaviours to form and maintain engaging social interactions with people. Cognitive Architectures emphasise a generality of mechanism and application, making them an ideal basis for such technical developments. Following the successful first workshop on Cognitive Architectures for HRI at the 2014 HRI conference, this second edition of the workshop focusses specifically on applications to social interaction. The full-day workshop is centred on participant contributions, and structured around a set of questions to provide a common basis of comparison between different assumptions, approaches, mechanisms, and architectures. These contributions will be used to support extensive and structured discussions, with the aim of facilitating the development and application of cognitive architectures to social HRI systems. By attending, we envisage that participants will gain insight into how the consideration of cognitive architectures complements the development of autonomous social robots.**

***Index Terms*—Cognitive Architectures; Cognitive Robotics; Social Human-Robot Interaction**

I. Introduction

Achieving social interactions between humans and robots is a complex task that has yet to be attained, but which is necessary for the increasing range of real-world applications for social robots. It requires an understanding of human social behaviour, and it requires the robots to use appropriate, adaptive and contingent behaviours to form and maintain these social interactions. Given that pre-programmed approaches are clearly insufficient for this problem, Cognitive Architectures provide a good alternative as they propose general mechanisms of 'intelligence' and behaviour generation.

Following the successful First Workshop on Cognitive Architectures for HRI held at the HRI conference in 2014 (Bielefeld) [1], we we are running a second edition, in which we focus specifically on cognitive architectures for *social* human-robot interaction[1]. As previously, the intention is to have the workshop be as inclusive as possible, catering both for experienced researchers in the area, but also for those for whom this may be a new topic. For all, we intend the workshop to provide a forum for discussion and the exchange of ideas. To facilitate this discussion and to provide a basis for a concrete contribution to the research community, we request short position paper contributions, and will organise a special issue (based on extended version of the position papers) after the workshop to consolidate the progress made, and provide a reference point for the community.

[1]https://sites.google.com/site/cogarch4socialhri2016/

Fig. 1. Workshop logo: cogs are typically used to represent cognition in an individual agent, this has been adapted to acknowledge the central role that interaction must play in social human-robot interactions in addition to the 'internal' cognition of the individuals.

II. Background

Cognitive Architectures are constructs (encompassing both theory and models) that seek to account for cognition (over multiple timescales) using a set of domain-general structures, mechanisms and/or processes [2]. Typically (but not necessarily) inspired by human cognition [3], the emphasis is on deriving a set of general principles of operation not constrained to a specific task or context. Despite the multitude of implementations used [4], they encourage the system designer to initially take a broader perspective than the computational mechanisms to be used and consider what sort of functionality needs to be present for the type of application, and how this relates to other cognitive competencies that are required.

For HRI, such an approach to building autonomous systems based on Cognitive Architecture would emphasise first those aspects of behaviour that are common across domains, before applying these to specific interaction contexts for evaluation. In the case of social interaction, the problems are numerous, encompassing the coordination of multiple sensory and motor modalities for the robot, the timing of proactive and reactive actions, and the recognition of interacting human states (cognitive, affective, physical, etc). Indeed, recent theoretical developments have emphasised the complex temporal coordination dynamics of human social behaviour, rather than the internal state of any individual agent [5]. This leads to questions regarding how the human should be taken into account in the action preparation/selection for the robot: explicit and individual models of performance, theory-of-mind, and/or generalised statistical models of human behaviour? It also gives rise to the question of whether and how the robot 'cognition' and actions should be directly informed by (or indeed constrained by) human psychology and physiology, with the complexity and 'non-optimal' behaviours that may result, e.g. [6]. Should our cognitive architectures for social robots be based directly on models of human behaviour, or is there no need for this? These, and related, questions are outstanding in the field and require addressing if the utility and efficacy of social robots in the real world is to be realised.

Up to now, there have only been limited and relatively isolated attempts to addressing these questions, particularly within the HRI community, with few examples of direct applications, e.g. [7]. Building on the first iteration of this workshop [1], we seek to bring together researchers who are attempting to formalise knowledge of appropriate robot behaviours for naturalistic interaction with people, typically emphasising generally applicable, holistic perspectives (i.e. striving to consider the full gamut of socially interactive behaviour rather than only individual aspects).

III. Outline of the Workshop

This workshop is aimed at researchers from a wide range of backgrounds who may be interested in applying concepts from Cognitive Architectures to their work, specifically Social HRI. Participation in this workshop is open to all interested researchers.

Prospective participants are requested to submit a 2-4 page position paper on (preferably) their work involving cognitive architectures (including the development and/or application thereof). In order to facilitate interactions and discussions at the workshop (by providing a basis for comparison), we ask that all authors additionally use their position papers to provide an answer to six guiding questions. These are as follows:

1) Why should you use cognitive architectures - how would they benefit your research as a theoretical framework, a tool and/or a methodology?
2) Should cognitive architectures for social interaction be inspired by and/or limited by models of human cognition?
3) What are the basic requirements for social interaction for a cognitive architecture?
4) How the requirements for social interaction would inform your choice of the fundamental computational structures of the architecture (e.g. symbolic, sub-symbolic, hybrid, ...)?
5) What is the primary outstanding challenge in developing and/or applying cognitive architectures to social HRI systems?
6) Can you devise a social interaction scenario that current cognitive architectures would likely fail, and why would this be the case?

Submission of a position paper is not a pre-requisite for attendance, and we encourage researchers to attend the workshop even if not willing/able to submit a position paper, in order to maximise community engagement and the uptake of these concepts within the field of HRI. By attending, we envisage that participants will gain insight into how the consideration of cognitive architectures complements the development of autonomous social robots.

IV. Organisers

Paul Baxter is a researcher at Plymouth University (UK) in the Centre for Robotics and Neural Systems, and the Cognition Institute. After obtaining a PhD in Developmental Cognitive Robotics (University of Reading, UK), he worked on the EU FP7 ALIZ-E project to apply and evaluate a memory-centred perspective on cognition to social child-robot interaction. His current research work involves the development of supervised autonomous therapy robots for children with ASD (EU FP7 DREAM project), with a specific focus on cognitive robot control.

Greg Trafton is a Cognitive Scientist at the Naval Research Laboratory in Washington, DC, USA. He has degrees in both Computer Science (Trinity University) and Psychology (Princeton University) and works on Human-Robot Interaction from a cognitive modeling / architectures perspective.

Séverin Lemaignan is a researcher at Plymouth University (UK) in the Centre for Robotics and Neural Systems, and the Cognition Institute, focusing on the cognitive pre-requisites of social interaction between humans and robots. He conducts both basic work on mechanisms like the Theory of Mind, and technical realisations on interactive robots.

Acknowledgement

This work was partially supported by the EU FP7 project DREAM (grant number 611391, http://dream2020.eu/), and the Marie Sklodowska-Curie Actions of the EU H2020 project DoRoThy (grant number 657227).

References

[1] P. Baxter and J. G. Trafton, "Cognitive Architectures for Human-Robot Interaction," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI '14*. Bielefeld, Germany: ACM Press, 2014, pp. 504–505.

[2] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research*, vol. 10, no. 2, pp. 141–160, jun 2009.

[3] R. Sun, L. A. Coward, and M. J. Zenzen, "On levels of cognitive modeling," *Philosophical Psychology*, vol. 18, no. 5, pp. 613–637, oct 2005.

[4] D. Vernon, *Artificial Cognitive Systems - A Primer*. MIT Press, 2014.

[5] E. Di Paolo and H. De Jaegher, "The Interactive Brain Hypothesis," *Frontiers in Human Neuroscience*, vol. 6, no. June, pp. 1–16, 2012.

[6] J. L. McClelland, "The Place of Modeling in Cognitive Science," *Topics in Cognitive Science*, vol. 1, no. 1, pp. 11–38, jan 2009.

[7] J. G. Trafton, L. M. Hiatt, A. M. Harrison, P. Tamborello, S. S. Khemlani, and A. C. Schultz, "ACT-R/E : An Embodied Cognitive Architecture for Human-Robot Interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 30–54, 2013.

Memory-Centred Cognitive Architectures for Robots Interacting Socially with Humans

Paul Baxter
Centre for Robotics and Neural Systems
The Cognition Institute
Plymouth University, U.K.
paul.baxter@plymouth.ac.uk

***Abstract*—The Memory-Centred Cognition perspective places an active association substrate at the heart of cognition, rather than as a passive adjunct. Consequently, it places prediction and priming on the basis of prior experience to be inherent and fundamental aspects of processing. Social interaction is taken here to minimally require contingent and co-adaptive behaviours from the interacting parties. In this contribution, I seek to show how the memory-centred cognition approach to cognitive architectures can provide an means of addressing these functions. A number of example implementations are briefly reviewed, particularly focusing on multi-modal alignment as a function of experience-based priming. While there is further refinement required to the theory, and implementations based thereon, this approach provides an interesting alternative perspective on the foundations of cognitive architectures to support robots engage in social interactions with humans.**

I. Introduction

The representation and handling of memory is an important feature of cognitive architectures, with a variety of symbolic and sub-symbolic representation schemes used (generally as passive storage), typically based on assumptions of modularity [1]. As such, memory is generally considered to be structurally separable from the cognitive processing mechanisms, and functions to provide these ‘cognitions’ with the required data.

In the memory-centred cognition perspective, memory is instead considered to be a fundamentally active process that underlies cognitive processing itself rather than being a passive adjunct [2], [3]. Based on evidence and models in neuropsychology, e.g. [4], this approach necessitates a re-examination of the organisation and functions of cognitive architectures, as outlined below (section III).

Previously, I put forward the case for the greater consideration of memory in HRI developments [5]. I argued that memory is pervasive: fundamentally involved in all aspects of social behaviour, beyond mere passive storage of information in data structures. In this brief (and relatively introspective) contribution, I expand on this point, exploring specifically the requirements of social interaction for robots, and consequently what cognitive architectures need to encompass.

II. Facets of Social Interaction

Social interaction is a complex phenomena that entails a range of abilities on the part of the interactants; indeed, there are facets of human-human social interaction that are as yet not fully understood, with the neural substrates supporting these in the individual yet to be characterised. One aspect that is commonly emphasised is the requirement for social signal processing for the individual, where behavioural cues (such as gaze, intonation, gesture, etc) should be interpreted to inform the behaviour of the observer.

One central idea emerging in the behavioural sciences is the notion of ’social contingency’: the coupling and co-dependency of behaviours between interacting individuals [6]. This explicitly acknowledges the necessary role that the ’other’ plays to set up the contingent behaviours, and moves away from the emphasis on social signal processing (though not discounting it). Minimal interaction paradigms provide intriguing illustrations of this: even given a low bandwidth interaction environment, there are non-trivial dynamics set up that cannot be explained by observations of an individual [7].

For social interaction generally, and in particular for this latter interacting systems perspective, there is an important role for prediction [8]. When interacting, there is an expectation that the interaction partner is also a social agent, and thus predicable in that context. Infants, for example, can use the gaze behaviour of a robot to infer that the robot is a psychological agent with which they can interact [9]. A previous study has further lent support to the idea that the imposition of expectations of social behaviour (and therefore the arising of socially contingent behaviours, in this case turn-taking) will come about if the interactants view each other as (potentially) social agents [10].

If the interaction partner (whether it is human or robot) is attributed with social agency, initially as a result of anthropomorphism for example [11], then one fundamental characteristic of social interaction between humans that will be seen is the ‘chameleon effect’ [12], or imitation/alignment, e.g. [13], [14], [15]. The presence of this within an interaction, as a type of contingency between the interactants (see above), could be seen as an indicator of sociality.

These phenomena, from attribution of social agency to alignment, illustrate a necessity for social robots (to a certain extent at least) to conform to human cognitive and behavioural features, as well as to their constraints, to enable predictability, consistency and contingency of robot behaviour with respect to the human(s) in the interaction.

III. Memory-Centred Cognitive Architecture

From neurospychology, the Network Memory framework [4] emphasises the central role that distributed associative cortical networks play in the organisation and implementation of cognitive processing in humans. The role of associative networks serves not only as a learning system (through Hebbian-like learning), but also as a substrate for activation dynamics. The reactivation and adaptation of existing networks combine to generate behaviour that is inherently based on prior experience.

The Memory-Centred Cognition perspective, as applied to the domain of cognitive robotics [2], seeks to extend these principles of operation: associative networks supporting activation dynamics that bring prior experience to bear on the current situation. A developmental perspective is necessary in order to do so [16]: the creation (and subsequent updating) of the associative networks must be done through the process of experience in order to form the appropriate associations between information in the present sensory and motor modalities of the robot (or system, in the case of a simulation).

Once an associative structure has been acquired, the principle mechanism at play is *priming* [2]. Priming in a memory-centred system occurs when some sub-set of the system is stimulated (from incoming sensory information for example), which causes activation to flow around the network, in turn causing parts of the network with no external stimulation to become active. Priming in this way fulfils a number of important functions. Firstly, it sets up cross-modal expectations, or the prediction of currently absent stimuli. Secondly, the priming process facilitates an integration of information across different modalities in a way that is explicitly based on prior experience (biased by the weights of the associative network).

A computational implementation of this has been applied to an account of the developmental acquisition of concepts [17]: not only was the system able to complete the task with a high success rate, but also the errors it made were consistent with those made by humans. A similar computational implementation has also been used to demonstrate how word labels for real-world objects can facilitate further cognitive processing [18]. These examples provide a glimpse of the range of cognitive processing (relevant to human cognitive processing) that can be accounted for using the memory-centred perspective.

Regarding social human-robot interaction, and in particular the notion that alignment is a fundamental feature of it (section II), the memory-centred perspective provides an intuitive, and indeed effective, account. Using exactly the same mechanism as for the concept learning study, the structure of an associative network was learned based on human behaviour (across a number of different modalities), which could then be directly used to determine the characteristics of the robot behaviour [14]. Alignment is achieved as a by-product of the way the memory-centred cognitive system operated: the associations were learned through experience, and behaviour was generated from priming (i.e. recall).

IV. Addressing Questions

From the context outlined above, I now attempt to provide answers to a set of six questions relevant to the notion of social cognitive architectures. I particularly seek to emphasise a principled-basis (as opposed to computational mechanism-basis) for cognitive architectures and for the application to social interaction.

A. Why should you use cognitive architectures - how would they benefit your research as a theoretical framework, a tool and/or a methodology?

The benefit would be in considering cognitive architectures as a set of principles (a theoretical framework), a methodology for assessing these principles, and as a tool for providing robots with autonomous intelligent behaviour.

There are in my view three specific contributions related to scientific development (as opposed to technical implementation) that cognitive architectures can make to HRI research and development, which are centred around the idea of a cognitive architecture being made up of a set of formalised hypotheses.

Firstly, in a principled manner, they allow data and theory from empirical human studies to be integrated into artificial systems. For example, if data from a psychology experiment is to be integrated, a framework for doing so is required (i.e. the architecture enables an interpretation of the data). This first point promotes the idea of a directly human-inspired/constrained architecture. Secondly, treating cognitive architectures as a set of formalised (through implementation) principles, they facilitate a comparison of different architectures at a level abstracted away from the computational systems/algorithms used, enabling a focus on the assumptions. In the presently considered case of social interaction, this is a useful facet given the as yet uncertain nature of what exactly constitutes social interaction (section II). Thirdly, the application of cognitive architectures (in robotic systems for instance) provides a means of evaluating its constituent assumptions and principles. This is related to the first point, but is focused more on the integration of empirical evidence obtained from application/experimentation with the architecture itself.

B. Should cognitive architectures for social interaction be inspired and/or limited by models of human cognition?

Following from the principles of social interaction outlined above, essentially, yes.

Taking the view that social interaction between humans is founded on the intrinsic tendency of humans to expect certain types of behaviour from their interaction partners (see section II), it becomes important to ensure that the robot will not violate expectations. In order not to violate expectation, there must necessarily be some understanding (either on the part of the system designer or learned by the system itself) of what expected human behaviour would be.

In the memory-centred cognition perspective, prior interaction history of the robot with humans would constrain its future behaviour by this experienced behaviour.

C. What are the functional requirements for a cognitive architecture to support social interaction?

The discussion of social interaction (section II) emphasised the importance of contingent behaviour, anticipation/prediction to support this, and adaptation/personalisation. In addition, it is necessary to specify appropriate timing, and embodiment-appropriate responses.

If socially-appropriate behaviour is in the eye of the (human) beholder, then the Keepon robot for example demonstrates the importance of coherence of behaviour and timing [19]. The minimally complex embodiment is convincingly responsive in a social manner, to the extent that it is seen as a communicative partner [20]. Even though it doesn't use language, only uses few degrees of freedom (in contrast to many other robots used in HRI), and is only minimally humanoid in appearance, the effect of apparent sociality is strong.

Integration of sensory and motor modalities in a temporally consistent and responsive manner (i.e. contingency), based on principles of prediction from prior experience (i.e. memory), and coherency with the robot embodiment used (c.f. Keepon example) are therefore fundamental functional requirements for a social cognitive architecture.

D. How would the requirements for social interaction inform your choice of the fundamental computational structures of the architecture (e.g. symbolic, sub-symbolic, hybrid, ...)?

Given the commitment to the memory-centred cognition perspective in this work, there is a natural fit with sub-symbolic computational structures. This provides a number of inherent advantages (section III), such as the integration of predictive behaviour from prior experience, and priming effects (within and between modalities).

However, the nature of applications in human-robot interaction (relying on language for example) means that it is not yet possible to dispense with symbol-processing systems. Nevertheless, there is in principle an effort to push the limits of sub-symbolic processing mechanisms up the processing and representation hierarchy, as revisited below (section V).

E. What is the primary outstanding challenge in developing and/or applying cognitive architectures to social HRI systems?

One of the primary challenges in the application of cognitive architectures to social interaction lies in the general lack of understanding of what is precisely involved in human-human social interaction. To a certain extent it is an attempt to find a solution to a problem that is as yet not fully characterised. This reflects on the requirements for the cognitive architectures that should engage in social interaction: if a commitment to human-like cognition/behaviour is made (see section IV-B), then what precisely are the constraints that need to be incorporated?

A more practical concern that requires further development is the provision of sensory systems for robots that can provide sufficiently complex characterisations of the (social) environment for effective decision making. There is however, in my opinion, no clear distinction between sensory systems and cognitive processing, given the necessity for interpretation of raw sensory signals (e.g. camera images) at various levels of abstraction.

F. Can you devise a social interaction scenario that current cognitive architectures would likely fail, and why?

The question is whether the application to a single domain can be generalised to other domains, which is where the benefits of cognitive architectures should come (section IV-A). As such, rather than a specific interaction scenario, I would suggest instead that autonomous sociality over variable time-scales poses challenges to current approaches and implementations.

In the short term, the challenge for social robots is to produce behaviour appropriate to the interaction context, informed by prior interaction experience, in a manner consistent with the expectations of the interacting humans. Furthermore, this socially interactive behaviour should adapt to the interaction partner over time, in terms of verbal and non-verbal behaviours for example. The technical challenges to support this in terms of sensory processing are outstanding, but there are also clear challenges in terms of the mechanisms of adaptation required (i.e. the 'cognitive' aspect). The memory-centred approach has ventured an implementation towards this problem, although the account is as yet incomplete.

Over extended periods of time, the challenges are compounded by requirements for stability. This is not just stability in terms of ensuring the system doesn't fail, but also in resolving the apparent trade-off between adaptability to new situations and robustness of the cognitive system. From the perspective of the memory-centred cognition account, the resolution to this question lies in how the formation, maintenance and manipulation of memory is handled in the system in terms of parameters and structures.

V. OUTLOOK

The nature of the discussion above is primarily principled and theoretical rather than focused on specific computational mechanisms. Naturally I believe memory-centred cognition perspective to have a consistency and coherence that merits consideration and further development. However, it is not in its current state able to practically support all aspects of real social interactions with real people.

This is a limitation shared with many 'emergent' cognitive architecture approaches [21]: theoretically interesting and coherent perhaps, but practically limited in terms of what can be done on real systems (use of language and dialogue being good examples of this). This is partly due to an implication of the theoretical perspective: by committing to a holistic approach that emphasises the integration and interplay of many different factors (including, for example, cognition, embodiment, culture, etc), the problem is made more difficult before a computational implementation is even begun. On a practical level, the types of dynamical system (be they neural network-based or other) used are typically not fully understood, or are at least highly complex [22], e.g. in terms of conditions for stability (particularly when adaptation/learning is incorporated), which does not bode well for social robots that have to be reliable in real interactions with real people.

For these reasons, I do not believe that symbol-based approaches should (or can) be discarded, at least not for the foreseeable future. They provide the means of getting closer to actually achieving the desired behaviours in reality. Having said this, and as noted above (sec. IV-D), I remain intent on pushing the boundary between symbolic and sub-symbolic implementations 'up' the abstraction hierarchy, in a manner common with a range of other developmentally-oriented researchers [23], [24].

So, what does a memory-centred cognitive architecture look like if it is to be effectively applied to social interaction? And what does the memory-centred cognitive architecture enable in terms of social robots that would be difficult to achieve with an alternative approach? The functionality of developmental learning of cross-modal associations for prediction and action generation outlined above (section III) provides a technically difficult but in principle effective solution to the issue of learning from a vast array of potential multi-modal information in a way that is useful for action generation. This is not to say that this is the only approach (theoretical or computational) that would be capable of a similar functionality. However, this is where the second aspect, the requirement to fulfill social interaction with humans through conformity with human cognition (section II), becomes a distinguishing characteristic of the memory-centred approach.

In developing the theory, I have applied it to a range of practical systems and applications, as reviewed above (section III). For example using the same mechanism, accounts have been made of concept acquisition [17] and multi-modal robot behaviour alignment to an interaction partner [14]. Other systems using the same principles have been used to demonstrate the development of low-level sensory-motor coordination through experience [16], and the role of words in supporting new cognitive capabilities [18].

Whereas my commitment to the memory-centred cognition perspective for robotics is strong, my commitment to the specific mechanisms used is weak. I must acknowledge that there are a number of weaknesses with the various systems used, notably related to hierarchical structure/representation, and an incomplete account of temporal processing. However, in my view, this does not invalidate the theoretical approach, and merely serves to provide motivation to either find or develop a more appropriate computational implementation that fulfils all of the principles and constraints of the memory-centred cognition perspective.

Acknowledgement

This work was supported by the EU FP7 project DREAM (grant number 611391, http://dream2020.eu/).

References

[1] R. Sun, "Desiderata for Cognitive Architectures," *Philosophical Psychology*, vol. 17, no. 3, pp. 341–373, sep 2004.

[2] P. Baxter, R. Wood, A. Morse, and T. Belpaeme, "Memory-Centred Architectures: Perspectives on Human-level Cognitive Competencies," in *Proceedings of the AAAI Fall 2011 symposium on Advances in Cognitive Systems*, Arlington, Virginia, U.S.A.: AAAI Press, 2011, pp. 26–33.

[3] R. Wood, P. Baxter, and T. Belpaeme, "A Review of long-term memory in natural and synthetic systems," *Adaptive Behavior*, vol. 20, no. 2, pp. 81–103, 2012.

[4] J. M. Fuster, "Network Memory," *Trends in Neurosciences*, vol. 20, no. 10, pp. 451–9, 1997.

[5] P. Baxter and T. Belpaeme, "Pervasive Memory: the Future of Long-Term Social HRI Lies in the Past," in *Third International Symposium on New Frontiers in Human-Robot Interaction at AISB 2014*, London, UK, 2014.

[6] E. Di Paolo and H. De Jaegher, "The Interactive Brain Hypothesis," *Frontiers in Human Neuroscience*, vol. 6, no. June, pp. 1–16, 2012.

[7] E. Di Paolo, M. Rohde, and H. Iizuka, "Sensitivity to social contingency or stability of interaction? Modelling the dynamics of perceptual crossing," *New Ideas in Psychology*, vol. 26, no. 2, pp. 278–294, 2008.

[8] E. C. Brown and M. Brüne, "The role of prediction in social neuroscience," *Frontiers in Human Neuroscience*, vol. 6, no. May, pp. 1–19, 2012.

[9] A. N. Meltzoff, R. Brooks, A. P. Shon, and R. P. N. Rao, ""Social" robots are psychological agents for infants: a test of gaze following." *Neural networks*, vol. 23, no. 8-9, pp. 966–72, 2010.

[10] P. Baxter, R. Wood, I. Baroni, J. Kennedy, M. Nalin, and T. Belpaeme, "Emergence of Turn-taking in Unstructured Child-Robot Social Interactions," in *HRI'13*, Tokyo, Japan: ACM Press, 2013, pp. 77–78.

[11] B. R. Duffy, "Anthropomorphism and the Social Robot," *Robotics and Autonomous Systems*, vol. 42, pp. 177–190, 2003.

[12] T. L. Chartrand and J. A. Bargh, "The Chameleon Effect: the perception-behavior link and social interaction," *Journal of Personality and Social Psychology*, vol. 76, no. 6, pp. 893–910, 1999.

[13] K. Dautenhahn and A. Billard, "Studying robot social cognition within a developmental psychology framework," in *Third European Workshop on Advanced Mobile Robots (Eurobot'99)*, Zurich, Switzerland, 1999, pp. 187–194.

[14] P. E. Baxter, J. de Greeff, and T. Belpaeme, "Cognitive architecture for humanrobot interaction: Towards behavioural alignment," *Biologically Inspired Cognitive Architectures*, vol. 6, pp. 30–39, 2013.

[15] A.-L. Vollmer, K. J. Rohlfing, B. Wrede, and A. Cangelosi, "Alignment to the Actions of a Robot," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 241–252, 2015.

[16] P. Baxter and W. Browne, "Memory as the substrate of cognition: a developmental cognitive robotics perspective," in *Proceedings of the Tenth International Conference on Epigenetic Robotics*, Örenäs Slott, Sweden, 2010, pp. 19–26.

[17] P. Baxter, J. D. Greeff, R. Wood, and T. Belpaeme, ""And what is a Seasnake?": Modelling the Acquisition of Concept Prototypes in a Developmental Framework," in *International Conference on Development and Learning and Epigenetic Robotics*. San Diego, USA: IEEE Press, 2012, pp. 1–6.

[18] A. F. Morse, P. Baxter, T. Belpaeme, L. B. Smith, and A. Cangelosi, "The Power of Words," in *Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*. Frankfurt am Main, Germany: IEEE Press, 2011, pp. 1–6.

[19] H. Kozima and C. Nakagawa, "Social Robots for Children: Practice in Communication-Care," in *AMC'06*. Istanbul, Turkey: IEEE Press, 2006, pp. 768–773.

[20] A. Peca, R. Simut, H.-L. Cao, and B. Vanderborght, "Do infants perceive the social robot Keepon as a communicative partner?" *Infant Behavior and Development*, vol. in press, 2015.

[21] D. Vernon, G. Metta, and G. Sandini, "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 151–180, 2007.

[22] R. D. Beer, "On the Dynamics of Small Continuous-Time Recurrent Neural Networks," *Adaptive Behavior*, vol. 3, no. 4, pp. 469–509, 1995.

[23] L. B. Smith, "Cognition as a dynamic system: principles from embodiment," *Developmental Review*, vol. 25, pp. 278–298, 2005.

[24] A. Cangelosi, *et al*, "Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 167–195, 2010.