Toward Autonomous Child-Robot Interaction: Development of an Interactive Architecture for the Humanoid Kaspar Robot

Abolfazl Zaraki
School of Computer Science
University of Hertfordshire
AL10 9AB Hatfield, UK
a.zaraki@herts.ac.uk

Kerstin Dautenhahn
School of Computer Science
University of Hertfordshire
AL10 9AB Hatfield, UK
k.dautenhahn@herts.ac.uk

Luke Wood
School of Computer Science
University of Hertfordshire
AL10 9AB Hatfield, UK
l.wood@herts.ac.uk

Ori Novanda
School of Computer Science
University of Hertfordshire
AL10 9AB Hatfield, UK
o.novanda@herts.ac.uk

Ben Robins
School of Computer Science
University of Hertfordshire
AL10 9AB Hatfield, UK
b.robins@herts.ac.uk

ABSTRACT

Recently, the development and deployment of social robots to support children with autism spectrum conditions (ASC) has attracted the attention of researchers in different fields. The main aim of this multidisciplinary research field is to teach and improve the social skills and capabilities of the children through interacting and playing with robots however, due to the spontaneous nature of children with ASC, as well as the inherent complexity of developing autonomous interactive robots, creating such systems is a challenging task. In this paper, we present our progress in the context of the BabyRobot project, towards developing an interactive humanoid robot to support autonomous child-robot interactions (CRI). We have been developing an interactive sense-think-act architecture to enable the humanoid robot Kaspar to display goal-oriented adaptive behaviours while playing, communicating, and collaborating with children. The final part of this paper reports on the initial tests of the proposed interactive architecture.

CCS Concepts

•**Human-centered computing** → *User interface toolkits;*

Keywords

Autonomous child-robot interaction; Autism spectrum condition; BabyRobot project; Interactive architecture;

1. INTRODUCTION

Children with ASC often have difficulty socially interacting with other people. For this reason, using social robots

Figure 1: A child with autism explores the correct distance to place the toy in front of the robot's eyes.

as a novel and engaging aid in the education and therapy of children with ASC has been proposed previously [1]. Many human-robot interaction (HRI) studies, have experimentally confirmed the usability and benefit of social robots for stimulating and teaching social skills in children with ASC [5]. For this reason, the development and deployment of social robots to support these children has increasingly attracted the attention of researchers. However, developing an autonomous robot is still a challenging task.

Recently, in the context of the BabyRobot project[1], we have studied the different developmental behavioural aspects of children with ASC, during their interaction with the Kaspar robot[2][2]. The main objective of our research activities is to focus on investigating the learning and communication development of the children, in dyadic and triadic interaction games, through defined enjoyable scenarios, where the children play different games with Kaspar. The first phase of the project has focused on designing and testing the scenarios that aim to assist in developing children's social skills and in particular, encourage them to improve their social collaborative skills to meet common goals. In this first phase, we have performed our CRI studies in a Wizard-of-Oz setup, for a usability evaluation, to allow further adaptation and

[1]www.babyrobot.eu/

[2]www.herts.ac.uk/kaspar

Child-Robot Interaction Workshop - HRI2017 , Vienna Austria

DOI: 10.1145/1235

tuning of the scenarios, as well as a technological assessment to establish the requirements for developing a socially interactive autonomous robot (see Figure 1). In the second phase, we will move towards performing similar studies in an autonomous/semi-autonomous mode by implementing the necessary technological developments to facilitate the interactions and autonomously control particular aspects of the robot's functions.

In this paper, we report our progress in developing an interactive architecture to enable autonomous CRI studies. Preliminary results indicate that the architecture of the system for autonomous CRI studies is particularly promising.

2. KASPAR HUMANOID ROBOT

Kaspar is a child-sized robot which functions as a platform for HRI studies, primarily utilising facial expressions and physical gestures (movements of the hand, arms, and gaze), to interact with humans. The robot is approximately 60 cm tall and sits in a fixed position (see Figure 4). The main body of the robot contains the electronic boards, batteries and servo motors. Kaspar has 18 FSR pressure sensors placed in several locations around the body including the hands, arms, shoulders, torso, head, legs and feet to detect tactile interaction. The robot has 11 degrees of freedom in the head and neck, 5 in each arm and 1 in the torso. The face is a silicon-rubber mask, which is supported on an aluminium frame. The robot's eyes are fitted with video cameras and can move up/down, and left/right, the eyelids can open and shut and the mouth is capable of opening, shutting, frowning and smiling. The Kaspar controller system has several pre-programmed behaviours that are typically used for generic play sessions and include various postures, hand waving, drumming on a tambourine that is placed on its legs and singing children rhymes [2].

3. THE INTERACTIVE ARCHITECTURE

In order to develop the interactive architecture (IA), we have been following the typical sense-think-act architecture [3] and implemented its components based on our robot's tasks and game scenarios defined in the project. Since one of the primary objectives in our studies is to use several toys to engage children to play with the robot, the first step towards creating an autonomous CRI, relies on Kaspar's competency to accurately recognize and track different toys as well as children's high-level communicative signals. These core features will assist Kaspar's ability to track the status of the game and adapt its behaviour accordingly to achieve the relevant educational or therapeutic objectives of the game. For this reason, in the IA development, we have defined the concept of 'object' as *something* with pre-defined characteristics and features, which plays an essential and known role in CRI. Following the same strategy, we have considered children, toys, environment, and the robot itself, as different objects, which have inherited different characteristics and features. For example, a person has been defined as an object with a head, skeleton, voice, and facial expression while a toy has been defined as an object with a name, colour, 3D position and orientation.

As shown in Figure 2, the IA has three main interconnected layers, which deal with Kaspar's *Sense* or perception, *Think* or decision making and action planning processes, and *Act* or low-level motor control and action performing capa-

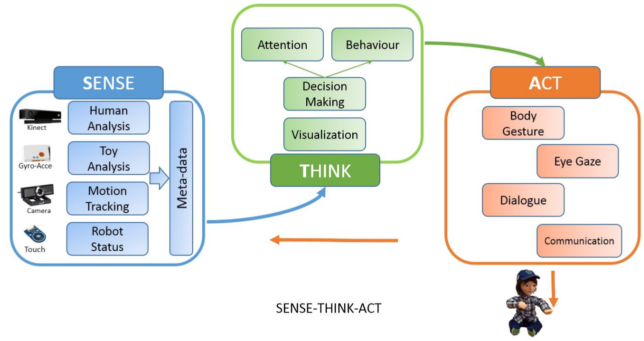

Figure 2: The Kaspar's interactive architecture for autonomous child-robot interaction.

bilities. The IA has fundamentally two functions, the first is to perceive the environment, using various sensors and devices, and the second is to control the robot's behaviour in an appropriate way in order to advance the game and close the interaction loop with children. IA functions as an active vision for the robot, which is aware of the interaction environment and adapts the robot's behaviour in real-time by taking into consideration the rules of the game, the tasks, as well as the child and game status. The following section briefly discusses the layers and their underlying components as well as the implementation process.

3.1 The Sense Layer

The input layer of IA acquires low-level sensory information through different devices and creates high-level perceptual information (meta-data) e.g., body gesture, which is understandable to the next layer. The sense layer collects the incoming data streams in real-time from a Kinect v2 device, a Gyro-Acc. sensor, a full-HD camera, and several touch sensors that have been placed on the robot's body. As shown in Figure 2, there are several parallel perceptual modules that have been integrated in order to track and analyse different *objects* e.g., children and toys. Each module is a software package that analyses a specific high-level feature of the objects, as follows:

3.1.1 Human Analysis Module (body analysis, speaker tracking, and speech recognition)

This perception task has been achieved by integrating the Microsoft Kinect Software Development Kit (SDK) v2.0 into the sense layer. It is able to simultaneously detect up to six humans in the robot's field of view (FOV) and track the 3D positions of 25 body joints for each person, in real-world coordinates. Using this information, the body gesture and head posture of the user can be recognized, which is important in CRI, due to their social communicative roles in human social interaction. The speaker tracking capability of the layer has been realised using the beam formation functions of the SDK. By comparing the beam angle in the environment with the 3D angles of the children seen in the FOV, the individual who speaks can be tracked in real-time. In order to recognize the speech of the speaker, we have integrated the Windows speech recognition platform into the sense layer. This module receives the audio signals through the Kinect's microphone array and recognizes the utterance of users based on a pre-defined list of the word/sentences that we have defined in the Kaspar's database. Using this module, the users will be able to verbally interact with the robot. However, the performance of the module as well as the recognition rate are still unknown for child-like voices, it performs relatively well with adult speech.

3.1.2 Toy Analysis Module (multiple toys tracking, toys 3D orientation estimation)

In order to analyse multiple toys, image processing techniques such as blob detection and colour filtering have been employed to detect and extract an object from the background and determine the pixel address in the 2D frame. For this reason, the object analysis module, firstly, acquires the image constructed by the RGB camera, and processes the image in order to convert its specifications (dimensions and pixel ratios) into the one required for the filtering step. The module then applies different filters to filter out the specific colours in order to identify the colour regions in the image. Finally, it returns as the output, the pixel address (x,y) of each object in the camera's FOV. In order to facilitate and improve the above aforementioned image processing tasks, the open source image processing/vision library called AForge has been integrated into the toy analysis module. However, due to the performance issues encountered, other image processing libraries such as Accord and EMGU Net will be evaluated in the future to assess their suitability for integration into the layer. This perception task performed by the following phases:

- Colour frame acquisition: the system acquires the colour frame of an external wide-angle camera, with up to 30 frame-per-second with the 1920 x 1080 pixel dimensions, and converts the image dimensions to 640x480. It then visualizes the new created colour frame in an image holder in the dedicated user interface. The UI allows users to view the camera's FOV and specify the target object for the system.
- Defining the target toys in the colour space: in this phase Kaspar operator should introduce the target toys for the system through the UI by a left-clicking with the mouse on the target object in the FOV. The system then picks the pixel address of the selected point and maps it into the camera's colour space. Since each pixel is presented by a mixture of the colour values, the system returns the pixel address (x,y) as well as the Red, Green, Blue (RGB) values of that pixel and passes these values to the colour filter.
- Filtering out specific colours: Once the RGB colour values have been determined, the user should specify the radius, which enables the system to filter out a range of colours instead of a specific colour (e.g., a range of the colour blue for a blue object). The centre of the range is the RGB values given by the operator and with the specified radius. The filter function, filters out the pixels of the image, and decides, which colour is inside of the RGB range and radius. Since the RGB values of the object might change in different lighting conditions, using the radius in the filter results in more robust colour tracking capabilities.
- Making a Threshold: In colour tracking, in some cases there are other unwanted objects which have similar colours to the target object. In order to remove the unintended objects, the Thresholding method can be used. This method converts the RGB frame to the grayscale and measures the light intensity of each pixel and applying a threshold, then returns 0 for the values of intensity lower than the threshold, and returns 1 for the values higher than the threshold. This phase results in a black and white image (binary image) in which only the selected object are shown and the background is removed.
- Blob Counter: once the background has been removed, the system has to localize the target toys and find the pixel address of the object in the image frame. For that, the system uses the blob counter technique, which takes advantage of a label algorithm to locate the toy. Once the object regions have been specified, we can apply an additional filter to filter out the toys based on their size or other feature such as shape.
- In the last phase the system overlays a rectangle on the selected toy, and comparing the RGB values of the pixel in centre of the rectangle with the robot database, it recognizes and targets the toy in the colour space.

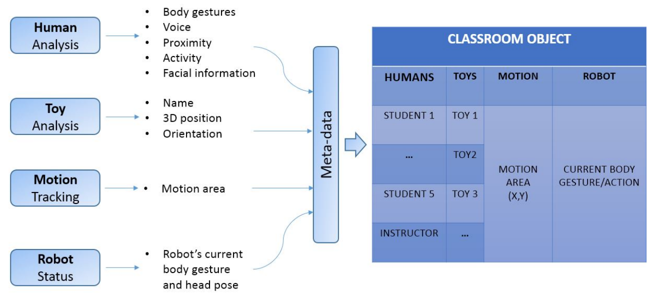

Figure 3: The structure of the meta-data in the sense layer.

In addition to the toy recognition and tracking capabilities, the module estimates the 3D orientation of a cube that we use in our game scenarios to assist the children in acquiring the skills being taught. This has been achieved by analysing the signals generated from the Gyro-accelerometer sensor, which has been placed inside of the cube. Analysing the raw data (acceleration signals, angular rate), the module returns three of the angles values (roll, pitch, yaw) of the cube. Knowing the 3D orientation of the cube is very important since this enables Kaspar to understand which side of the cube is currently being observed by the child and which side is being presented to the robot. To ensure the precision of the module in obtaining the 3D orientation of the cube, the output of the orientation estimation algorithm has been connected to a virtual cube in which we can control its motion by moving the physical cube (see Figure 5).

Once the required perceptual information has been collected by the sense layer it then creates the meta-data, which is temporary memory to store the information about the interaction environment (see Figure 3). The system manages the data based on the concept of the 'object' that we have defined. The sense layer creates the meta-data following the strategy presented in [7]. As shown, there are different data categories in which each of them manages and stores the data in several sub-categories. For example, as shown in Figure 6, the 'objects' presented in the FOV are one person and three toys (two toys + one cube). Based on this data, the sense layer returns the high-level data of the 'objects', and stores the data of person in the STUDENT 1, and also stores the toys information in the TOY1, TOY2, and TOY3 data holders respectively.

Figure 4: Some of the Kaspar's body gestures and eye gazes generated by the act layer.

3.2 The Think Layer

Once the meta-data has been created and received by the think layer, the layer firstly visualizes the high-level data on the screen. Then, in a decision making process it identifies the region of interest (ROI) (attention target), which can be for example the face of a child, a toy, or the motion area, and decides how the robot should react to the information presented in that ROI. The attention mechanism behind the decision making process is based on the game scenarios and the tasks that robot should follow to achieve the pre-defined goal. Currently for the decision making process we are using a rule-based mechanism however, we are planning to integrate IrisTK [6] to the think layer in order to have a better control on the robot's behaviour and reactions in a multi-party CRI.

3.3 The Act Layer

Once the information (ROI/attention target, and the robot's behaviour/reaction) has been identified, the think layer sends this information to the act layer through a dedicated YARP port [4]. The robot's controller machine in the act layer receives the data and generates the required control signals for robot's actuators in order to display correct behaviour on the robot. As shown in Figure 4, the act layer is able to display different body gestures and head poses for the robot. The data communication between the robot controller machine and Kaspar can be performed either through a wired or wireless (Wi-Fi) network.

4. THE PRELIMINARY TEST OF IA

We evaluated our interactive architecture in a human-robot interaction experiment with Kaspar. The purpose of this experiment was twofold: (i) to demonstrate if each individual layer and the modules are able to detect and track the specific features in real-time, and (ii) to demonstrate if the overall IA is able to manage the data flow from the sensory level to the robot control level in order to enable Kaspar to shape an autonomous social interaction with a human. For this reason, we defined a HRI scenario in which the subject was instructed to initiate a social interaction with Kaspar and introduce two new toys to Kaspar, which are presented in different colours (red and blue). The subject then repositioned the toys and asked the Kaspar to see if it could find and track the toys in the FOV. As shown in Figures 5 and 6, the IA was able to detect and analyse different high-level features that shows each of the sense, think, and act layer as well as the underlying modules were working correctly. Furthermore, the preliminary evaluation of the autonomous interaction between Kaspar and the person demonstrated a promising capability of the proposed architecture in enabling an autonomous interaction with Kaspar.

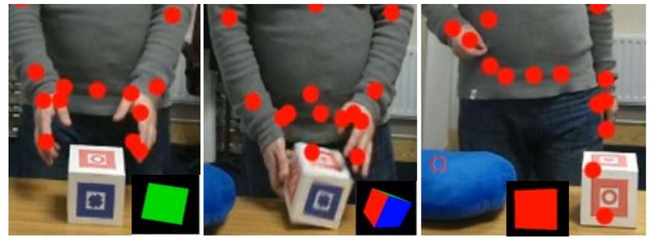

Figure 5: Cube orientation tracking using a Gyro-Accelerometer sensor.

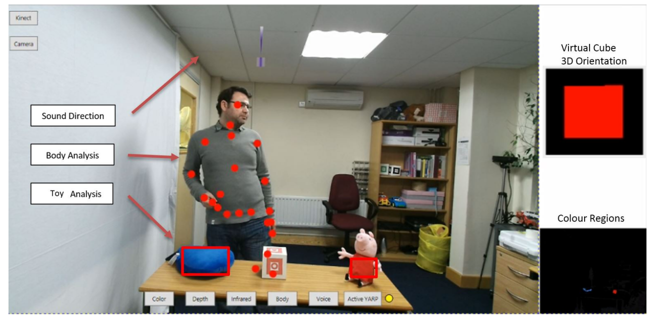

Figure 6: The visualisation of high-level features on the Kinect and camera's images.

5. ACKNOWLEDGMENTS

This work has been partially funded by the BabyRobot project supported by the EU Horizon 2020 Programme under grant 687831.

6. REFERENCES

[1] H.-L. Cao, C. Pop, R. Simut, R. Furnemónt, A. De Beir, G. Van de Perre, P. G. Esteban, D. Lefeber, and B. Vanderborght. Probolino: a portable low-cost social device for home-based autism therapy. In *International Conference on Social Robotics*, pages 93–102. Springer, 2015.

[2] K. Dautenhahn, C. L. Nehaniv, M. L. Walters, B. Robins, H. Kose-Bagci, N. A. Mirza, and M. Blow. Kaspar–a minimally expressive humanoid robot for human–robot interaction research. *Applied Bionics and Biomechanics*, 6(3-4):369–397, 2009.

[3] E. Gat et al. On three-layer architectures. *Artificial intelligence and mobile robots*, 195:210, 1998.

[4] G. Metta, P. Fitzpatrick, and L. Natale. Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*, 3(1):8, 2006.

[5] P. Pennisi, A. Tonacci, G. Tartarisco, L. Billeci, L. Ruta, S. Gangemi, and G. Pioggia. Autism and social robotics: A systematic review. *Autism Research*, 2015.

[6] G. Skantze and S. Al Moubayed. Iristk: a statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 69–76. ACM, 2012.

[7] A. Zaraki, M. Pieroni, D. De Rossi, D. Mazzei, R. Garofalo, L. Cominelli, and M. B. Dehkordi. Design and evaluation of a unique social perception system for human-robot interaction. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–15, 2016.