Computing and Informatics, Vol. 40, 2021, 83–103, doi: 10.31577/cai_2021_1_83

MULTI-PLATFORM INTELLIGENT SYSTEM FOR MULTIMODAL HUMAN-COMPUTER INTERACTION

Mateusz Jarosz, Piotr Nawrocki, Bartłomiej Śnieżyński

Institute of Computer Science
Faculty of Computer Science, Electronics and Telecommunications
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: {mateusja, piotr.nawrocki, bartlomiej.sniezynski}@agh.edu.pl

Bipin Indurkhya

Institute of Philosophy
Jagiellonian University
Golebia 24, 31-007 Krakow, Poland
e-mail: bipin.indurkhya@uj.edu.pl

Abstract. We present a flexible human–robot interaction architecture that incorporates emotions and moods to provide a natural experience for humans. To determine the emotional state of the user, information representing eye gaze and facial expression is combined with other contextual information such as whether the user is asking questions or has been quiet for some time. Subsequently, an appropriate robot behaviour is selected from a multi-path scenario. This architecture can be easily adapted to interactions with non-embodied robots such as avatars on a mobile device or a PC. We present the outcome of evaluating an implementation of our proposed architecture as a whole, and also of its modules for detecting emotions and questions. Results are promising and provide a basis for further development.

Keywords: Human–computer interaction, multi-platform, intelligent system architecture, multimodal system, humanoid robot

1 INTRODUCTION

Social humanoid robots are becoming more and more commonplace. Their capabilities are also increasing rapidly: they are equipped with a variety of sensors to obtain information about the surrounding environment, including people, along with a variety of mechanisms to perform human-like actions naturally. Therefore, the importance of the Human–Robot Interaction (HRI) research is increasing.

There is a perceived need for a system that could be used during HRI experiments. Commercial manufacturers provide software frameworks for programming their respective robots and executing code (e.g. NAOqi). Systems developed by researchers are also becoming available [18, 20]. The problem is that such systems are usually closed, engineered to operate on a specific type of robot, and cannot be deployed on mobile devices.

One of the goals of our research is to improve the level of human–robot interaction (HRI) and design social robots with which humans can interact intuitively. To achieve this, we have designed and implemented an architecture that allows flexible interaction between humans and robots, which is the main contribution of this paper. In this architecture, the emotional state and the mood of the user are sensed on the basis of the users' dialogue with the robot, their posture and gesture, speech prosody, facial expression and eye gaze. An appropriate behaviour is then selected on the basis of a multi-path scenario. This architecture can be easily adapted to interactions with non-embodied robots such as avatars on mobile devices or PCs. Here, we describe our proposed architecture and report on a series of evaluation experiments. We also compare our architecture with other similar systems.

This article is structured as follows. Section 2 discusses related work. Section 3 contains a high-level description and implementation details of our proposed architecture. In Section 4 we describe evaluation experiments and their results. Section 5 compares our architecture with other similar approaches, while concluding remarks and directions for further research are presented in Section 6.

2 RELATED WORK

In recent years, many human–robot interaction architectures have been proposed based on behaviour trees, multimodal systems and adaptive systems. In this section we briefly describe some of these systems, along with their advantages and disadvantages. We focus on behaviour planning, decision-making using behaviour trees, and benefits of following a multimodal approach.

Alonso-Martin et al. [16] propose a multimodal emotion detection system as part of a larger Human–Robot Interaction system. It uses two channels of emotion detection, namely voice and face video, which are combined into one emotion value. A dialogue system is driven by this emotion value, acknowledging the intended effect

on the user. Our proposed architecture uses a similar approach, but we also incorporate information gleaned from the ongoing human–robot dialogue to determine the emotional state of the user.

In an earlier work, Breazeal [17], using her expressive anthropomorphic robot *Kismet*, studied emotions and expressive behaviours in regulating social interaction between a human and a robot in communicative and teaching scenarios. In this work, models of humanoid robot emotions and their scientific basis are described, and adapted for implementation in *Kismet*. They also use the prosody of the user's voice to detect their emotional stance. In the current prototype of our architecture we rely on face video for emotion detection, but we also plan to incorporate speech prosody in future versions. Moreover, at the moment we work with a non-expressive humanoid robot *Pepper*, but we plan to use expressive humanoid robots such as *Little Einstein* in the future.

More recently, Coronado et al. [20] proposed a robot programming framework and an interface for the development of usable and flexible end-user applications. The framework employs a component-based methodology, a block- and web-based interface, and a behaviour tree approach to designing robot behaviour, all of which can be combined to adopt the end-user development paradigm. This system is easy to use from the end-user perspective, and cross-platform tools like ROS and ZeroMQ are provided to enable the creation of platform-independent applications. It can also be expanded with new sensory devices or robots. Our architecture shares the multi-platform approach with the platform of Coronado et al., but the usability of our interface will be addressed in the future versions.

Beer et al. [18] developed a framework for Levels of Robot Autonomy (LORA), ranging from teleoperation (non-autonomous) to fully autonomous. Their framework proposes a 10-point taxonomy for LORA, and relates it to three HRI variables, namely acceptance, situation awareness, and reliability. However, compared to our architecture, this work focuses mostly on autonomous robot operation.

A human–robot interaction framework that outlines a general structure of future home service robots to assist humans in their home-based daily activities was proposed by Lee et al. [19]. The authors describe three main interaction modules: multimodal, cognitive, and emotional. The main function of the multi-modal interaction module is to make the interaction intuitive for the human user. The cognitive interaction module facilitates cooperative sharing of tasks, while the emotional interaction module maintains a close relationship between the human and the robot. Our framework is also multimodal in terms of accepting inputs from various kinds of sensors. We furthermore provide a cognition/perception module to infer higher-level conceptual information from the basic input devices. For example, we can extract the human's emotional state from a video feed (see Subsection 4.2), as long as the subject remains visible.

Ardila et al. implemented an adaptive controller for a robot arm [2], which can adapt motion trajectories to the environment and an overall robot interaction profile. This adaptive controller uses the PAD emotional model (Pleasure, Arousal and Dominance), where PAD values are used to change the strategy of robot movements.

This system generates affective motions in non-humanoid robots for more intuitive human–robot interaction. In our approach, we focus on adjusting behaviours instead of movements and our solution is capable of working on PCs, mobile devices and humanoid robots.

Rincon et al. [3] propose a novel cognitive-robot control architecture to adapt robot actions and motions to the dynamics of both the environment and the human. This solution involves incorporating "expressive states" in a cognitive model that adapt to yield optimal robot control. The authors also provide deep-learning algorithms for perception, cognitive models based on affects, and adaptive generalized predictive controllers (AGPC). Their system also relies on the PAD concept to represent the robot's emotional state. Adaptation is controlled by an AGPC, which changes according to the cognitive state of the robot. The AGPC cost functions are calculated using PAD values. An evaluation of the system showed that the robot was able to perform tasks continuously with expressive and personalized behaviours. This research focused on non-humanoid robots and adaptation of robot movements, whereas our system is targeted for PCs, mobile devices and humanoid robots, and we focus on adjusting behaviours as a whole rather than partially.

A framework based on an adaptive predictive control scheme and a fast dynamic and geometric identification process was proposed by Hagane et al. [4]. The approach was demonstrated with a force-controlled wall-painting task performed by a lightweight robot called KUKA. This research also includes a comparative analysis of the performance of generalized predictive control (GPC), adaptive proportional derivative gravity compensation, and adaptive GPC (AGPC). The results revealed that predictive controllers are more suitable than adaptive PD controllers with gravitational compensation, owing to the use of well-identified geometric and inertial parameters. This work also focused on non-humanoid robots and on performing movements adapted to the changing environment. In contrast, our research focuses on adapting more general behaviours, and is targeted not only for robots, but also for PCs and mobile devices.

Abiyev et al. proposed a novel behaviour tree (BT)-based control for decision-making in robot soccer [5]. The robot analyzes the current world state and decides how to act. The BT approach allows modelling of complicated situations with ease, which constitutes an advantage of this technique over finite state machines, which are widely used in robot control. An evaluation of the system performed by the authors reveals that BT performs well at the task of playing robot soccer. Though the use of BT in this system is similar to our approach, Abiyev's work is more focused on movements and goal-oriented decision-making, whereas our research is focused on adapting the behaviour flow in a multi-platform system.

Marzinotto et al. [6] proposed a unified BT framework along with notions of equivalence between BTs and Controlled Hybrid Dynamical Systems. They also demonstrate the applicability of their framework to real systems by scheduling open-loop actions in a grasping mission for the Nao robot. Their proposal to use BT for movement control is interesting, but differs from our approach in that we use a structure similar to BT for making decisions in a scenario tree.

Arriaga et al. propose a novel approach to emotion and gender recognition using only camera for human–robot interaction systems [11]. This approach uses a convolutional neural network (CNN) based on the simplified Google Xception model architecture. The system, as implemented by the authors, achieved 96 % accurancy in the IMDB gender dataset and 66 % in the FER-2013 emotion dataset, whereas humans achieve an emotion detection accuracy of 65 % ± 5 % in the same dataset, with the best solution peaking at 71 %. A major advantage of this model is its low computational cost, which is achieved by cutting the number of network parameters from 600 000 in the naive CNN implementation to 60 000, which corresponds to a tenfold reduction compared to their initial naive implementation, and 80-fold compared to the original CNN. This improvement enables the robot to run both networks at the same time and obtain results in real time. In our work, we rely on similar concepts for network design, but retrain the network with a larger data set, and change the face detection method to make it faster and avoid frames with undetected faces in high-framerate feeds.

Question detection in human–robot interaction is important: even if the robot cannot answer the question, it should react to it in some way. Ando et al. [12] propose a novel approach for question detection using lexical cues in addition to acoustic data. They also proposed their own framework for training the network, called feature-wise pre-training, which combines acoustic and phonetic features effectively. Their system achieved 66.8 % precision and 62.8 % recall for question detection. These are remarkable results, but we nevertheless decided to use CNN and mel spectrograms based on acoustic event detection for faster execution using the simplified Xception model.

Inception modules in conventional neural networks can be interpreted as an intermediate step between regular convolution and depthwise separable convolution operation (depthwise convolution followed by pointwise convolution). In this light, depthwise separable convolution can be understood as an Inception module with a maximally large number of towers. This observation led Chollet [13] to propose a novel deep convolutional neural network architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions of an Xception architecture. This architecture achieved slightly better results on the imageNet dataset than the inception model. In our work, we use a simplified version of the Xception model (fewer layers) for emotion and question detection.

Zhang et al. described an approach to sound event detection using conventional neural networks [14]. Conventionally, sound event recognition methods based on informative front-end features such as MFCC, or with back-end sequencing methods such as HMM, tend to perform poorly in the presence of interfering acoustic noise. As noise corruption is usually unavoidable, Zhang et al. proposed to use CNN and spectrograms as a more robust solution. This method achieved excellent performance under noise-corrupted conditions compared to the conventional state-of-the-art approaches in standard evaluation tasks. In our research, we apply the same basic idea to recognise questions, treating the last two-thirds as a sound event.

Perzanowski et al. presented a multimodal system for human–robot interaction [7] using three different sources of commands: speech, gestures and PDA interaction. They assumed that with an integrated system, the user will be less concerned with the means of communicating, and can therefore concentrate on the tasks and goals at hand. Though this research is old, it still incorporates basic emotions and demonstrates the advantages of a multimodal approach. We apply the same principle in designing our system.

Many research papers have been published in the area of human–robot interaction, adaptive movement or movement of robot parts, including the benefits of behaviour trees and multimodal systems. Among the important directions of HRI research is detection and analysis of emotions with the use of voice and face image. An important aspect, which, as indicated by a study of the available literature, has not been further explored, is the possibility of extending analysis of emotions to include information obtained from the dialogue between a robot and a human, as well as data from various types of sensors (for example, determining the robot's position). Detecting emotions, including, primarily, image analysis performed for this purpose, may exploit machine learning methods such as CNN. It is important that these methods are trained on as large a data set as possible in order to obtain the best possible results. As a result, the emotional attitude of the user may be accurately determined. In parallel, an important conclusion from the presented analysis is the need for universality of the developed solutions. Many of the systems analyzed in this section are designed only for a specific device or class of devices. Evidently, there is no existing multimodal human-computer interaction system that incorporates – whether fully or partially – behaviour trees for adaptation of behaviours, and is capable of operating on multiple platforms (such as robots and mobile devices). This is precisely what motivated us to initiate the research presented in this paper.

3 ARCHITECTURE FOR MULTIMODAL HUMAN-COMPUTER INTERACTION

Our architecture for multimodal human-computer interaction (see Figure 1) consists of the following modules: Perception modules, Actions, Activity scripts and Behaviour planner. Activity scripts represent scenarios, and are stored in the JSON format. These scripts are used by the Behaviour planner to enact a scenario. The Behaviour planner is the main module responsible for executing activity scripts and choosing appropriate paths in the script according to data from perception modules. Perception modules are responsible for extracting higher-level data about the user's interaction with the robot, based on raw data from the robot's sensors. In the future, we plan to augment this with external sensors. In the current prototype, there are four perception modules: facial expression, gaze tracking, dialogue monitoring, and speech prosody. The action module is responsible for executing simple actions, such as saying something, moving forward, changing the direction of the robot's gaze,

making hand gestures, or displaying a video. Different actions can be combined for more complex expressive movements.

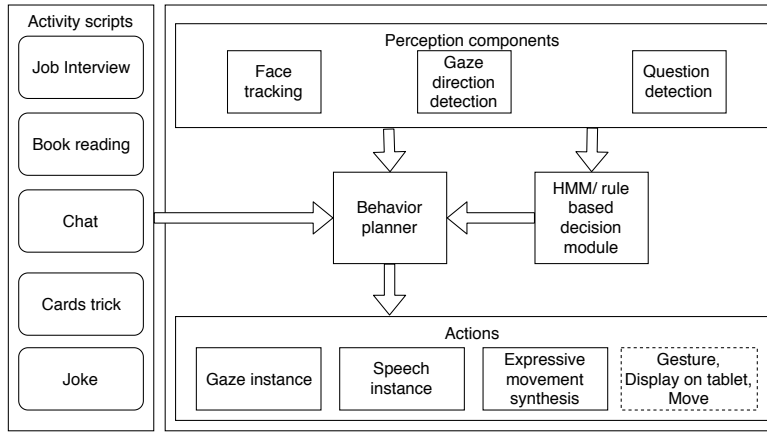

Figure 1. Framework architecture

Our proposed architecture has three main features:

Multi-platform: We designed our architecture to be able to work with different robotic platforms. At the moment, we mostly work with the Pepper robot, but we are planning to use other robots, such as Little Einstein, and also non-embodied avatars running on computers or mobile devices. To incorporate this multi-platform capability, we designed our architecture in such a way that only the part responsible for robot/avatar actions and extracting sensory data is dependent on the platform. All other parts of the system can be executed on any platform equipped with a Python interpreter.

Multimodality: Obtaining information from multiple sensors and analysing it simultaneously can result in significant computing load, especially for a mobile device with limited resources. To address this issue we decided to move most of the analysis tasks to the cloud as microservices. This approach has another benefit in that different platforms can share the same functionality through the cloud to avoid duplication. Moreover, as new features are added to the cloud, they become available across all platforms. In this way, we were able to deploy e.g. emotion and question detection facilities.

Intelligent system: Through a number of in-the-wild studies on child-robot interaction [21, 22], we have found that in order for a robot to behave naturally towards a human it needs to have a basic understanding of what humans are saying and where they are looking, along with some awareness of the surrounding environment. For example, the robot should be aware when a human is asking it something and should answer appropriately. We plan to achieve this

by using a multi-path scenario and an event system with break points. This is explained in detail below.

Initially, we tried a simple approach, shown on the left in Figure 2, where there was only one scenario, with the robot executing one behaviour. Clearly, this is not very flexible, for the robot should be able to take different actions and execute different behaviours depending on the context and user actions. To address this issue, we created a module containing several activity scripts. These were organised in a tree structure, with each node equipped with a condition, which determines when the given path should be followed, and a sequence of actions (e.g. speak, move, etc.) that the robot will perform while traversing the node. There are break points between actions where events can be executed. Each event consist of a type and a series of actions. Activity scripts also allow parallel looped paths for executing continuous or regularly repeating actions during a scenario. We can express our scenario structure formally in the following way:

$$scenario = (N, E, A, C, Ne, s) \tag{1}$$

where N is a set of nodes, and E is a set of edges. Nodes contain one or more actions from set A, a condition from set C and events from the set of events Ne: for all $n \in N$, node n is labeled by (ne, a, c), where $ne \in Ne$, $a \in A$, $c \in C$. Every event $ne \in Ne$ is labeled by action and condition: $(a, c) \in A \times C$. Every condition assumes the form of K-SAT:

$$\left(x_1^1 \wedge x_1^2 \wedge \cdots \wedge x_1^{n_1}\right) \vee \left(x_2^1 \wedge x_2^2 \wedge \cdots \wedge x_2^{n_2}\right) \vee \cdots \vee \left(x_k^1 \wedge x_k^2 \wedge \cdots \wedge x_k^{n_k}\right) \tag{2}$$

where x_i^j is a basic formula of the form $a == | <= | >= |! = | > | < b$, a and b are constant values or variables representing object states, results of detecting humans or emotions, etc. Ne are events defined per node. Every event has a set of actions and a triggering condition. *Scenario* has one starting node $s \in N$ and can have many leaf nodes and cycles.

Algorithm 1 shows how action scripts work. The MAIN procedure contains the main decision loop, where we browse the graph for child nodes, evaluating the condition and then choosing the best node to follow, e.g. the first one for which the condition evaluates to true. In addition to sequential nodes, we also identify parallel ones. Such nodes are run in separate threads and they perform repeating tasks with little to no impact on the main flow of the script, e.g. moving the robot's head in a repeating pattern. Parallel nodes can be stopped at any time, or terminate themselves, but are usually stopped at the end of the script. The next step involves execution of nodes (RUN procedure), with an event mechanism that operates as follows. Before taking any action, we register, in a decision module, all events defined for the given node. Event conditions are monitored in a parallel thread (see the PROCESS_EVENTS procedure). If a condition evaluates to true, a corresponding event is executed. Execution is synchronised with the main flow of scenario by locking a mutex which corresponds to the node. Actions are then performed sequen-

tially. While this goes on, a decision module can decide to trigger – based on data from perception components – one of the registered events. Such triggered events can start immediately, if the currently running action permits this, or they can be executed after the current action has ended. After executing all actions, the events are deregistered, and will not be called unless registered again.

Algorithm 1 Behavior planner algorithm

```
procedure MAIN
    graph ← read_node_graph()
    node ← get_current_node(graph)                          ▷ get root node
    while graph.has_next() do                    ▷ if current node has children
        graph.set_current_node(node)
        RUN(node)
        successors ← get_successors(node)
        pss ← get_passing_sequential_successors(successors)
        pps ← get_passing_parallel_successors(successors)
        for each parallel_successor ∈ pps do
            start_thread(parallel_successor)
        node ← get_best_successor(pss)
procedure RUN(node)
    if check_node_start_condition(node) == True then
        for each event ∈ get_events(node) do
            add_event_to_event_monitoring_list(event)
        Execute in a new thread
          PROCESS_EVENTS(event_monitoring_list, thread)
        for each action ∈ get_actions(node) do
            lock.acquire                ▷ Check if some event is not triggered
            action.run()
            lock.release
        for each event ∈ get_events(node) do
            remove_event_from_event_list(event)
        Stop thread
procedure PROCESS_EVENTS(event_monitoring_list, thread)
    while thread is running do
        for each event ∈ event_monitoring_list do
            if check_event_condition(event) == True then
                lock.acquire
                execute_action(event)
                lock.release
```

This architecture is loosely based on the concept of a behaviour tree [6]. Though behaviour trees have a formal definition and are compact, they are difficult to use for new users due to their non-intuitive structure. In contrast, directional graphs with

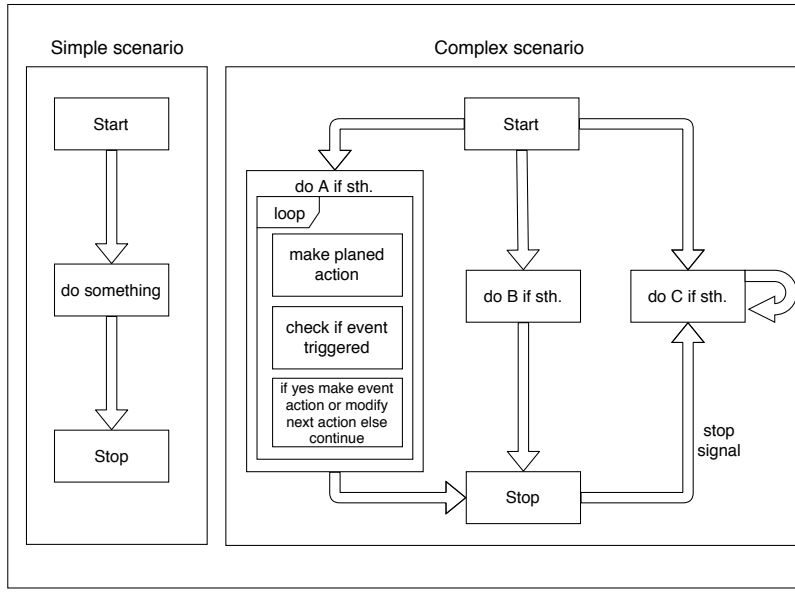

Figure 2. Comparison between a simple scenario and a complex scenario

entry conditions are natural for computer scientists and can be readily used with myriads of algorithms. Therefore, our representation is similar to a state machine or a classical directional graph with conditional nodes.

In order to improve performance and to incorporate a multi-platform approach, all the heavy computing perception modules, except the basic question detection module, were implemented as microservices. In our previous research [10], we tested several approaches to detecting the user's gaze direction with a camera mounted on a robot or a mobile device. Based on these results, we decided to use the OpenFace library [8] for estimating the user's gaze direction. Our solution also employs a system we developed ourselves for determining the user's gaze direction based on facial landmark [9] and pupil detection.

A simple approach to detecting questions is based on the assumption that the final second of a question exhibits higher pitch then the preceding several seconds. This solution is simple enough to be computed locally rather than by uploading the recording to the network, which might result in higher costs and increased complexity. Therefore, this module was not implemented as a microservice.

A more complex approach would involve text-based question detection based on a transcript returned by Google Cloud, followed by a dialog-act classification. This method was not implemented in our framework, because initial tests revealed high latency and cumulative error introduced by two main components: Polish speech-to-text conversion and dialog-act classification. Other important issues included the cost of using Google Cloud services, privacy issues (uploading the voice feed

to Google Cloud) and the difficulty of implementing continuous detection (Google Cloud limits streaming duration to 1 minute). Consequently, we decided to use dialog-act classification using Mel Spectrograms of length 3 s, with one-second overlap, as input for a CNN classifier. The CNN architecture is based on the Xception network with five Xception blocks and 200 000 parameters: we found that this works well for rapid computation on relatively slow devices.

The training data set for the CNN was prepared as follows. First, we partitioned audiobooks into distinct sentences and questions. All pieces containing more than one sentence were then removed. Finally, we augmented the sound samples by cropping 10 % of the data randomly, increasing volume by a factor between 2 and 5 (randomly), masking 10 % of possible data and replacing it with 0 s, adding 0.03 of random noise, using VTLP (Vocal Tract Length Perturbation), changing pitch, and varying speed by a factor between 0.5 and 2.0. For all data files, no augmentation was applied in the first 20 % and the final 20 % of the audio file (as in [23]), and we took the last 3 seconds of the data to extract its spectrogram. The training data set was in Polish and consisted of 10 500 samples (3 600 before augmentation), equally divided into questions and non-questions. The testing data set contained 7 500 samples with 1 150 questions (20 % of questions in the training data set prior to augmentation), similar to real-life dialogue.

For emotion detection, we retrained the network used in [11] by adding one Xception block and extending the training set threefold, with additional 100 000 images from "The Ryerson Audio-Visual Database data set" [15]. We also found that using CNN to find the face in an image, instead of using a hog classifier, results in a significant improvement in the face detection rate.

The main module for the robot is written in Python, and consists of the following sub-modules: The Action-script reader reads an action script and returns a directional graph with one root and many possible paths to the end. The Configuration manager is responsible for preparing the system environment. The Behaviour engine is responsible for executing the graph produced by the Action-script reader. It decides which of the multiple paths is to be followed based on data from the Analysis module. The Analysis module collects data from the perception modules. It can perform local processing to further perform statistical analysis of the raw sensory data robot and/or combine data from different sensors synchronously or asynchronously, using sensor fusion algorithms. This module is also responsible for triggering events based on the collected/analysed data. Actions are primitive operations stored as objects inside the graph produced by the Action-script reader.

4 EVALUATION

To evaluate the efficacy of our architecture, especially with regard to emotion and question detection modules, we conducted an experiment along with a number of field studies. Below, we describe these tests and present their results.

4.1 Verification of Scenario Execution

The first experiment involved an interview setting: the Pepper robot was the interviewer and a human participant was the interviewee. The robot asked a number of predetermined questions, and listened to responses from the interviewee. Figure 3 shows results before and after experiment surveys. In the aftermath of the interview, subjects claimed to be less concerned about the robot.

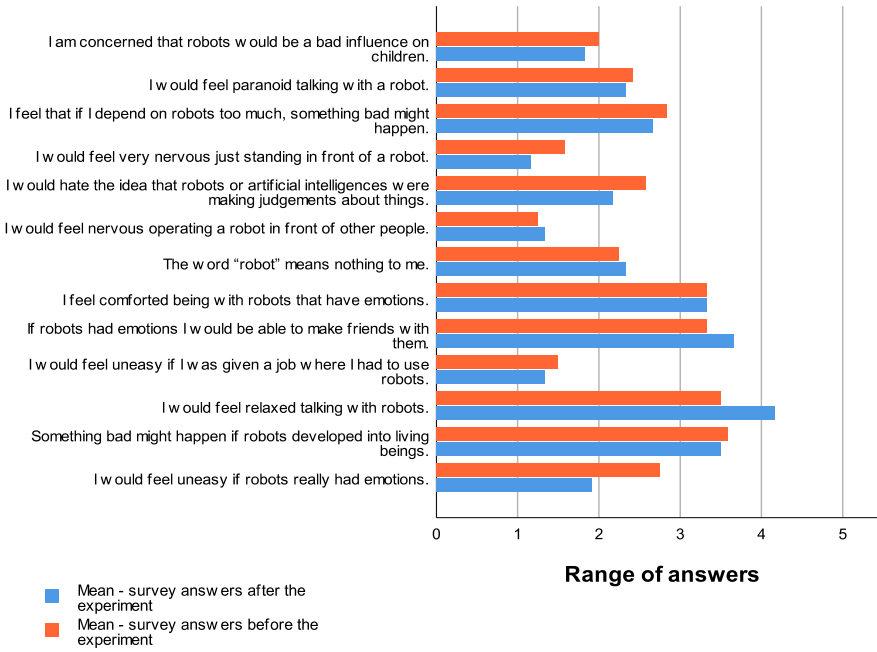

Figure 3. Results before and after experiment survey (1)

Subsequently, we conducted a number of field studies (in-the-wild studies) in which the Pepper robot interacted with children. Here, we discuss results from two such studies. The first study involved kindergarten children (age 4–6) at a Polish school in Kraków. The interaction scenarios were designed to match the children's capabilities, and included a drawing activity, a reading session (Pepper reads to the children), dancing with the robot, and a question-answer session using the Wizard-of-Oz paradigm (meaning that a human experimenter answered the children's questions that were delivered through Pepper).

We conducted another such workshop with older children (age 5–13), where we introduced a rock-paper-scissors game with Pepper, along with an extended question-answer session. All experiments proceeded without any problems affecting either the software or the robot, and were well received by the children.

The presented experiments were performed using a simplified version of our software (without emotion recognition and question detection); thus, to confirm that the system works well when equipped with new modules, we conducted a small study with the same setup as in the interview experiment, including emotion recognition and question detection. Participants gave positive responses when robot reacted to their behaviours. The questionnaires filled out by participants before and after the interview (see Figure 4), reveal an increase in the level of anxiety. This was due to the robot's reaction to the behaviour of the participants, e.g. having detected that a participant was sad, the robot asked if everything was okay. Such interruptions could have prevented the participant from answering a previously asked question, which made the participant feel a bit confused. This minor problem will be addressed in a future version of our system. There were no other problems. The system performed well and participants did not report any other issues.

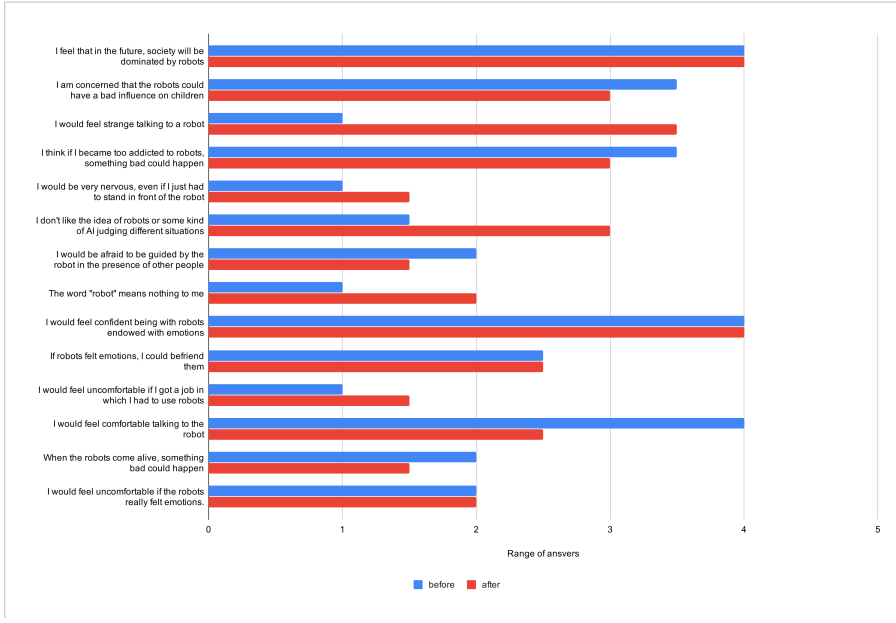

Figure 4. Results before and after experiment survey (2)

4.2 Evaluation of the Emotion Classification Module

As mentioned above, we used a modification of the network proposed in [11] and retrained it on a larger dataset. We also changed the face detection classifier from the haar classifier in the OpenCV library to the CNN classifier in the dlib library. The main reason for this change was better performance of the CNN classifier compared to the haar classifier. In our tests, haar dropped many frames without detecting

a face even though the face was clearly visible: in one experiment, the haar classifier found the face in only 4 of 81 frames in one file, whereas CNN recognised the face in 75 of 81 frames. The average rate of frames with undetected faces was 15 % when using haar, but drooped to 8 % when using the CNN classifier.

For improving the efficacy of emotion detection we retrained our model on a larger dataset. We added one additional Xception block to the network to increase the model's capacity, and added face scans extracted from the Ryerson Audio-Visual Database [15] to the FER-2013 dataset. This caused an unbalance in the dataset due to the lack of faces displaying disgust or surprise in the Ryerson Audio-Visual Database. Figure 5 shows the confusion matrix of the retrained neural network on the test dataset: we can observe that while results are generally promising, disgust detection has a lower accuracy compared to other emotions like anger and sadness.

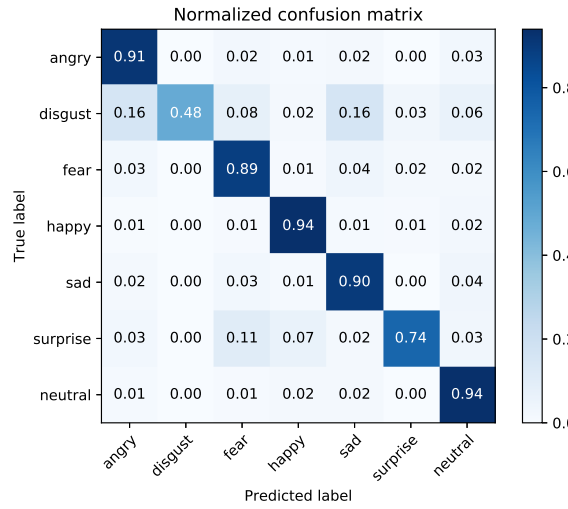

Figure 5. Confusion matrix on the emotion test dataset

The average accuracy of the model is 86 % with 83 % recall.

4.3 Evaluation of Question Discovery Module

First we attempted a simple approach, resulting in peak accuracy of 70 % for one class and 30 % for the other class, time or 55 % for both classes combined. A text-based approach was abandoned after preliminary testing. An evaluation (confusion matrix) of the question detection module using the CNN and Mel spectrograms is shown in Figure 6.

On average, our model achieved 70 % precision and 80 % recall due to an unbalanced test data set; however, when weighted with the number of samples, the results improved to 88 % precision and 82 % recall.

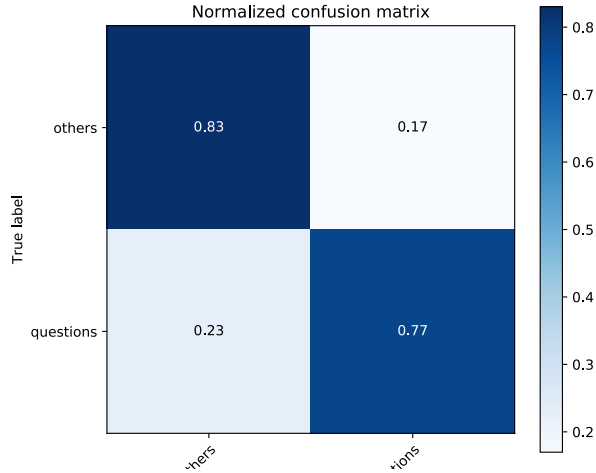

Figure 6. Confusion matrix on audio test dataset

5 COMPARISON WITH OTHER FRAMEWORKS

In this section we compare the main features of our architecture with other existing systems for HRI. We selected external systems on the basis of the following criteria: functional similarity, creation in the last 7 years (most are 2-3 years old) and popularity (most have more then 10 citations). Other important factors included a clear description of the system and a similar range of applications. We could not test all other systems ourselves, as all information we have is based on published papers and therefore the number of features which can be compared is limited to those mentioned in such papers. We chose six features which are common to all systems and mentioned in their respective publications. The first and second features, "Target environment" and "Compatible robots and environments", correspond to one of our main goals: multi-platform operation. As the next feature we chose "Understanding emotions" – we believe that recognising human emotion and exploiting that information in the course of conversation is important and represents the future of HRI, similar to "Question detection". Next, we focused on the presence of a GUI as it can greatly lower the skill threshold for interaction scenarios and is important for that reason. The final two features, "Allows modification of scenario according to changing environment" and "Decision algorithm" correspond to our other goals, namely adaptivity and "intelligence" of the system. Some systems include other important features, such as gesture recognition, but those features are only present in a minority of the analyzed solutions, so we decided to omit them.

Table 1 briefly summarizes the main features of five popular HRI systems, as well as of our solution.

	Our solution	Alonso-Martin [16]	Ardila [2]	Coronado [20]	Röning [24]	Liu [25, 26]
Target environment	Multi-platform	Multi-platform	Single-platform	Multi-platform	Single-platform	Multi-platform
Compatible robots and environments	Pepper/nao family, virtual avatar on mobile and PC	Pioneer robot/ROS compatible	Robo arm	Nao	Minotaurus	Nao, mobile robot
Understanding emotions	Seven discrete emotions for better interaction	Seven discrete emotions for better interaction	PAD (Pleasure, Arousal, and Dominance) for modifying movements	None	Six discrete emotions for better interaction	Six discrete emotions for better interaction; also the robot can show emotions
Question detection	Yes	No	No	No	Yes	No
GUI	No	No	No	Yes	No	No
Allows modification of scenario according to changing environment	Yes	Yes	Yes	Yes	Yes	Yes
Decision algorithm	Conditional Tree	Rules	Robust Generalised Predictive Control	Behaviour trees	N/A	N/A

Table 1. Comparison of HRI system features

All systems enable making adjustments in the scenario depending on changes in the environment, for example user emotions. Our system, as well as the systems described by Alonso-Martin, Röning and Liu use discrete emotion representation, while the Ardila system uses a continuous pleasure arousal model. Both approaches have advantages – in our opinion, discrete representation is simpler and easier to understand. Moreover, in the Ardila system emotions mean the state of the robot, while in other systems they express the state of the human in the environment, and the Liu system not only recognizes human emotions, but can also display robot emotions. Only the Coronado system does not employ an emotion detection module. We think that emotions are one of most important channels in human communication; what is more, humans tend to be more open when talking with a robot, showing their emotions freely. Due to those facts, we believe that an emotion detection module is an essential component of an HRI system. Our system includes a question detection module; Alonso-Martin and Coronado systems can use one, if available, due to their modular architecture. The question detection module enables more natural conversation between the human and the robot, and is also simpler and less demanding then a speech-to-text solution with an advanced chatbot to interpret and respond to user speech. The Röning system uses such an online chatbot to answer user questions. Our system can be run on most platforms and can interact not only with robots, but also with virtual agents. Alonso-Martin and Liu systems are also compatible with many robots.

Our system uses conditional trees whereas the Alonso-Martin system uses rules, the Adrila system relies on a special algorithm, the Coronado system uses behaviour trees and the remaining two systems use an unknown decision mechanism. In our opinion our approach is equivalent to the rule-based approach in terms of simplicity, and incorporates the advantages of the hierarchical approach, such as behaviour trees. Only the last system has a graphical interface, but we plan to create one for our system in the near future. From this simple comparison we can see that our solution is needed because its features are not replicated by other freely available systems. What is more, our system is more flexible than other solutions since it is able to operate on most platforms and to be started on many systems.

6 CONCLUSIONS

Our goal was to create an architecture for human–robot interaction which is intuitive and incorporates the emotional state of the user. We developed an approach based on behaviour trees for controlling the flow of interaction. To evaluate our architecture, we implemented a prototype system and conducted a number of experiments in varying conditions. We also carried out a detailed comparison of our system with other similar systems. The results demonstrate the flexibility of our architecture, which allows a robot to react to human questions in an appropriate way. We also achieved good performance on the test data set with our recognition module.

The architecture is universal (can be applied in many scenarios), distributed and heterogeneous: less demanding services can be located on the robot platform or mobile device, while more complex ones (e.g. using neural networks to process images or signals) can be offloaded to the public cloud or a local PC. This enables efficient processing of multi-modal data. Owing to the microservice approach, the system can be adapted to work in other environments (e.g. Virtual Machines instead of the cloud and PC) and new hardware platforms (other robots or VR avatars). Currently a single control module (Behaviour planner) is responsible for executing a synchronous main scenario, represented by a graph, and processing asynchronous events in parallel. In the future, this approach can be scaled up and several instances of the Behaviour planner, controlling different robots, can be deployed.

In the future we also plan to conduct more extensive tests of the system and generate an improved version based on user feedback.

Acknowledgements

The research presented in this paper was supported by funds from the Polish Ministry of Science and Higher Education allocated to the AGH University of Science and Technology. Mateusz Jarosz's work was supported in part by the National Center for Research and Development (NCBR) under Grant No. POLTUR2/5/2018. We wish to thank Anna Kolota for her assistance with testing.

REFERENCES

[1] Fine, S.—Singer, Y.—Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications. Machine Learning, Vol. 32, 1998, No. 1, pp. 41–62, doi: 10.1023/A:1007469218079.

[2] Ardila, L. R.—Coronado, E.—Hendra, H.—Phan, J.—Zainalkefli, Z.—Venture, G.: Adaptive Fuzzy and Predictive Controllers for Expressive Robot Arm Movement During Human and Environment Interaction. International Journal of Mechanical Engineering and Robotics Research, Vol. 8, 2019, No. 2, pp. 207–219, doi: 10.18178/ijmerr.8.2.207-219.

[3] Rincon, L.—Coronado, E.—Law, C.—Venture, G.: Adaptive Cognitive Robot Using Dynamic Perception with Fast Deep-Learning and Adaptive On-Line Predictive Control. In: Uhl, T. (Ed.): Advances in Mechanism and Machine Science (IFToMM WC 2019). Springer, Cham, Mechanisms and Machine Science, Vol. 73, 2019, pp. 2429–2438, doi: 10.1007/978-3-030-20131-9_240.

[4] Hagane, S.—Ardila, L. K. R.—Katsumata, T.—Bonnet, V.—Fraisse, P.—Venture, G.: Adaptive Generalized Predictive Controller and Cartesian Force Control for Robot Arm Using Dynamics and Geometric Identification. Journal of Robotics and Mechatronics, Vol. 30, 2018, No. 6, pp. 927–942, doi: 10.20965/jrm.2018.p0927.

[5] Abiyev, R. H.—Akkaya, N.—Aytac, E.: Control of Soccer Robots Using Behaviour Trees. 2013 9th Asian Control Conference (ASCC), IEEE, Istanbul, Turkey, 2013, pp. 1–6, doi: 10.1109/ascc.2013.6606326.

[6] Marzinotto, A.—Colledanchise, M.—Smith, C.—Ögren, P.: Towards a Unified Behavior Trees Framework for Robot Control. 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 5420–5427, doi: 10.1109/icra.2014.6907656.

[7] Perzanowski, D.—Schultz, A. C.—Adams, W.—Marsh, E.—Bugajska, M.: Building a Multimodal Human–Robot Interface. IEEE Intelligent Systems, Vol. 16, 2001, No. 1, pp. 16–21, doi: 10.1109/mis.2001.1183338.

[8] Wood, E.—Baltrusaitis, T.—Zhang, X.—Sugano, Y.—Robinson, P.—Bulling, A.: Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 3756–3764, doi: 10.1109/iccv.2015.428.

[9] Baltrusaitis, T.—Robinson, P.—Morency, L.-P.: Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild. Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2013, pp. 354–361, doi: 10.1109/iccvw.2013.54.

[10] Jarosz, M.—Nawrocki, P.—Placzkiewicz, L.—Sniezynski, B.—Zielinski, M.—Indurkhya, B.: Detecting Gaze Direction Using Robot-Mounted and Mobile-Device Cameras. Computer Science, Vol. 20, 2019, No. 4, doi: 10.7494/csci.2019.20.4.3435.

[11] Arriaga, O.—Valdenegro-Toro, M.—Plöger, P.: Real-Time Convolutional Neural Networks for Emotion and Gender Classification. 2017, arXiv: 1710.07557.

[12] Ando, A.—Asakawa, R.—Masumura, R.—Kamiyama, H.—Kobashikawa, S.—Aono, Y.: Automatic Question Detection from Acoustic and Phonetic Features Using Feature-Wise Pre-Training. Proceedings of INTERSPEECH, 2018, pp. 1731–1735, doi: 10.21437/interspeech.2018-1755.

[13] Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807, doi: 10.1109/cvpr.2017.195.

[14] Zhang, H.—McLoughlin, I.—Song, Y.: Robust Sound Event Recognition Using Convolutional Neural Networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 559–563, doi: 10.1109/icassp.2015.7178031.

[15] Livingstone, S. R.—Russo, F. A.: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English. PLoS ONE, Vol. 13, 2018, No. 5, Art. No. e0196391, doi: 10.1371/journal.pone.0196391.

[16] Alonso-Martin, F.—Malfaz, M.—Sequeira, J.—Gorostiza, J. F.—Salichs, M. A.: A Multimodal Emotion Detection System During Human–Robot Interaction. Sensors, Vol. 13, 2013, No. 11, pp. 15549–15581, doi: 10.3390/s131115549.

[17] Breazeal, C.: Emotion and Sociable Humanoid Robots. International Journal of Human-Computer Studies, Vol. 59, 2003, No. 1-2, pp. 119–155, doi: 10.1016/s1071-5819(03)00018-1.

[18] Beer, J. M.—Fisk, A. D.—Rogers, W. A.: Toward a Framework for Levels of Robot Autonomy in Human–Robot Interaction. Journal of Human–Robot Interaction, Vol. 3, 2014, No. 2, pp. 74–99, doi: 10.5898/jhri.3.2.beer.

[19] Lee, K. W.—Kim, H. R.—Yoon, W. C.—Yoon, Y. S.—Kwon, D. S.: Designing a Human–Robot Interaction Framework for Home Service Robot. IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2005), 2005, pp. 286–293, doi: 10.1109/ROMAN.2005.1513793.

[20] Coronado, E.—Mastrogiovanni, F.—Venture, G.: Development of Intelligent Behaviors for Social Robots via User-Friendly and Modular Programming Tools. 2018 IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO), Genova, Italy, 2018, pp. 62–68, doi: 10.1109/arso.2018.8625839.

[21] Zguda, P.—Kołota, A.—Jarosz, M.—Sondej, F.—Izui, T.—Dziok, M.—Belowska, A.—Jędras, W.—Venture, G.—Śnieżynski, B.—Indurkhya, B.: On the Role of Trust in Child-Robot Interaction. 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 2019, pp. 1–6, doi: 10.1109/ro-man46459.2019.8956400.

[22] Zguda, P.—Kołota, A.—Jarosz, M.—Sondej, F.—Izui, T.—Dziok, M.—Indurkhya, B.: "Why Don't You Have a Wife?!" Free Format Dialogue in CRI. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Late Breaking Results Poster, IEEE, 2019.

[23] Edward, M.: NLP Augmentation. Available at: `https://github.com/makcedward/nlpaug`, 2020.

[24] Röning, J.—Holappa, J.—Kellokumpu, V.—Tikanmäki, A.—Pietikäinen, M.: Minotaurus: A System for Affective Human–Robot Interaction in Smart Environments. Cognitive Computation, Vol. 6, 2014, No. 4, pp. 940–953, doi: 10.1007/s12559-014-9285-9.

[25] Liu, Z.—Wu, M.—Cao, W.—Chen, L.—Xu, J.—Zhang, R.—Mao, J.: A Facial Expression Emotion Recognition Based Human–Robot Interaction System. IEEE/CAA Journal of Automatica Sinica, Vol. 4, 2017, No. 4, pp. 668–676, doi: 10.1109/jas.2017.7510622.

[26] Liu, Z. T.—Pan, F. F.—Wu, M.—Cao, W. H.—Chen, L. F.—Xu, J. P.—Zhang, R.—Zhou, M. T.: A Multimodal Emotional Communication Based Humans–Robots Interaction System. 2016 35th Chinese Control Conference (CCC), IEEE, 2016, pp. 6363–6368, doi: 10.1109/chicc.2016.7554357.

Mateusz Jarosz is Ph.D. student in the Institute of Computer Science at the AGH University of Science and Technology, Krakow, Poland. His research interests include human–robot interaction, gaze patterns and machine learning. He is currently working on an NCBR-supported research project in cooperation with Turkish scientists.

Piotr Nawrocki is Associate Professor in the Institute of Computer Science at the AGH University of Science and Technology, Krakow, Poland. His research interests include distributed systems, mobile systems, cloud computing, artificial intelligence and service-oriented architectures. He has participated in several EU research projects including MECCANO, 6WINIT and UniversAAL. He is a member of the Polish Information Processing Society (PTI).

Bartłomiej Śnieżyński received his Ph.D. degree in computer science in 2004 from the AGH University of Science and Technology, Krakow, Poland. In 2004, he worked as Postdoctoral Fellow under the supervision of Professor R. S. Michalski at the Machine Learning and Inference Laboratory, George Mason University, Fairfax, VA, USA. Currently, he is Associate Professor in the Institute of Computer Science at AGH. His research interests include machine learning, multi-agent systems and knowledge engineering. He is a member of the Polish Information Processing Society (PTI) and the Polish Artificial Intelligence Society (PSSI).

Bipin Indurkhya is Professor of cognitive science at the Jagiellonian University, Krakow, Poland. His main research interests are social robotics, usability engineering, affective computing and creativity. He received his Master's degree in electronics engineering from the Philips International Institute, Eindhoven (The Netherlands) in 1981, and his Ph.D. in computer science from the University of Massachusetts at Amherst in 1985. He has taught at various universities in the US, Japan, India, Germany and Poland; and has led national and international research projects with collaborations from companies like Xerox and Samsung.