Adapting an hybrid behavior-based architecture with episodic memory to different humanoid robots

François Ferland, Arturo Cruz-Maya and Adriana Tapus
Robotics and Computer Vision Laboratory
ENSTA-ParisTech
Palaiseau, France
Email: {firstname.lastname}@ensta-paristech.fr

Abstract— **A common goal of robot control architecture designers is to create systems that are sufficiently generic to be adapted to different robot hardware. Beyond code re-use from a software engineering standpoint, having a common architecture could lead to long-term experiments spanning multiple robots and research groups. This paper presents a first step toward this goal with HBBA, a Hybrid Behavior-Based Architecture first developed on the IRL-1 humanoid robot and integrating an Adaptive Resonance Theory-based episodic memory (EM-ART). This paper presents the first step of the adaptation of this architecture to two different robots, a Meka M-1 and a NAO from Aldebaran, with a simple scenario involving learning and sharing objects' information between both robots. The experiment shows that episodes recorded as sequences of people and objects presented to one robot can be recalled in the future on either robot, enabling event anticipation and sharing of past experiences.**

I. INTRODUCTION

In autonomous human-interacting robots, the role of a control architecture is to orchestrate the various perception and action modules in a coherent manner to produce a meaningful and natural behavior. In artificial intelligence, the term *cognitive architecture* is generally used to describe the infrastructure supporting an intelligent system [8]. Autonomous robots can be described as *embodied* intelligent systems *situated* in their environment, as they perceive their environment directly instead of dealing with abstractions [1].

Depending on their level of abstraction, there are many ways to describe such architectures. As presented by Hawes and Wyatt [6], they can be described according to three layers: as an *computational architecture*, where a structure to process information is described without a specific problem in mind; as an *instantiated information-processing architecture*, where the structure of the previous level can be applied to a specific problem; and as a *software architecture*, or how the structure of the second level can be implemented in concrete terms.

One common goal of robot software architects is to build software systems that can both solve different problems and be compatible with multiple robots. However, this has both technical and logistics challenges. Firstly, different robots can have different software requirements, even when popular development environments such as ROS [12] can help smoothing this process. Secondly, it is not always common for a single research group to have access to multiple, different robots.

As such, the link between an architecture solving human-robot interaction problems, its underlying software framework, and the robots it is meant for, is an interesting subject of study. This corresponds mostly to the third layer of abstraction of Hawes and Wyatt. Recently, work has been done on applying model-based software engineering approaches to robotics system design, aiming to facilitate a smooth transition from problem space to operational space [13], [14]. The CoSy Architecture Schema (CAS) [5] and its toolkit (CAST) is an example of an architecture and underlying infrastructure that has been used on more than robot [7]. Another architecture that evolved to support multiple robots is the Motivational Behavioral Architecture (MBA) [11]. Its successor, the Hybrid Behavior-based Architecture (HBBA), used the same basic structure with a different, ROS-based infrastructure and adapted it to the IRL-1 humanoid robot [4].

HBBA integrates an episodic memory (EM) using Adaptive Resonance Theory (ART) neural networks [2], [16]. Collecting information about one's experiences over time and their relationships within a spatio-temporal context is a role associated to an episodic memory model [17]. Wang et al. [20] demonstrated the use of a cascade of two ART networks to create their electromagnetic adaptive resonance theory (EM-ART) model: one network encode spatial events, and the other extracts the temporal pattern of events to create episodes. The EM found in HBBA is based on this EM-ART model, and its implementation was validated within an object delivery context [9].

Before continuing in this direction, it is important to validate if the architecture along with its EM implementation can be transferred to other robots. In this paper, an instance of HBBA is adapted to two different robots: the Meka M-1 and the NAO from Aldebaran. Our goal is to validate if it is still possible to record and recall episodes based on a sequence of interaction events on robots that are significantly different from the one that saw the development of the architecture. A simple scenario where participants sit in front of a robot and show it various sequences of objects has been designed. These learned sequences of objects are tested to see if the robots can correctly anticipate the incoming objects. Furthermore, sequences recorded on one robot are also tested for recall on the second robot to see if the memory can be shared between two robotic embodiment of the architecture.

This is a first validation step before using the architecture in more complex interaction situations.

The paper is organized as follows. Section II briefly presents the host robots for the architecture. Section III describes the control architecture put in place. Section IV focuses on how objects' learning was performed, along with how sequences of events were presented to both robots. Section V presents and discusses the results from the experiment, and finally Section VI concludes the paper and proposes future directions on the usage of the architecture.

II. ROBOT HARDWARE

The experiment presented in this work has been conducted with a Meka M-1 robot and a NAO robot from Aldebaran. The Meka M-1, shown in Fig. 1, is a wheeled humanoid robot that has been designed to work in human-centered environments. The robot features compliant force control throughout its body, durable and strong hands, and an omnidirectional base with a prismatic lift. The head is a 7 Degrees-of-Freedom (DoF) robotic active vision head with high resolution FireWire cameras in each eye, integrated DSP controllers, and zero-backlash Harmonic Drive gearheads in the neck. Designed for a wide range of expressive postures, it is a platform particularly well suited for researchers interested in human-robot interaction and social robotics. Two computers are embedded in the mobile base of the robot: one for real-time tasks such as motor control, and one for higher level tasks such as signal processing and behavior coordination.

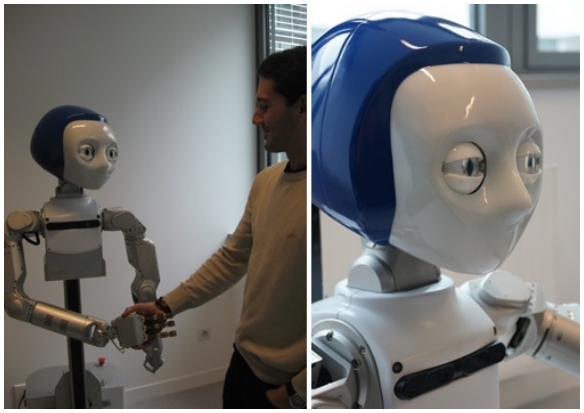


Fig. 1: The Meka M-1 robot

The NAO robot from the Aldebaran Robotics is a 53 centimeters tall bipedal humanoid robot with up to 25 DoF. Its head integrates two high resolution color video cameras, along with microphones and a speaker to conduct spoken interaction. Parts of its shell feature capacitive tactile sensors and sonar rangefinders. Its wide availability and ease of use make it a popular research platform in human-robot interaction.

III. CONTROL ARCHITECTURE

As its name implies, HBBA is a a hybrid, multilevel architecture corresponding to the "Think and Act Concurrently" paradigm [10]. Its current implementation is a collection of independent ROS nodes, written in C++ and Python. Source for non robot- or experiment-specific modules is available online[1].

Figure 2 illustrates an instance of HBBA for the experiment presented in this paper, along with robot-specific modules. For a more compact representation, configurations for both robots are merged in a single figure. However, the actual instance of HBBA for the Meka robot does not include modules for the NAO robot, and vice versa for the NAO instance. All modules of the architecture are meant to be executed on a single computer embedded on each robot. However, the architecture can also be distributed on multiple computers. This capability was exploited in this experiment. To accelerate the sharing of the content of the EM-ART between both robots, all non-robot specific modules were executed on a separate laptop computer and communicated with the other modules through ROS and an Ethernet network. This enabled us to simply switch between ROS master servers that are running on the robots, instead of copying data from one robot to the other between experiment phases. For other situations, the implementation of the EM-ART module saves its complete state as a file after each modification, but can also transmit it through ROS at runtime for monitoring or mirroring between two sites.

A. Sensors and Actuators

The only sensors used in this experiment, shown in light blue (see Figure 2), are a single camera for each robot. On the NAO robot, this camera corresponds to the one situated above the eyes of the robot. This camera provides a 640x480 RGB picture at a rate of 15 frames per second. It has an horizontal field-of-view (HFOV) of $60.97°$. On the Meka robot, the camera integrated in its right eye is used. This camera is configured to provide a 800x600 RGB picture, also at a rate of 15 frames per second. It has an HFOV of $31.50°$.

As for the actuators, shown in light yellow (see Figure 2), only the respective speaker of each robot is accessed for producing speech.

B. Perception Modules

Perception modules are shown in medium blue in Figure 2. Object Recognition is based on FindObject2D[2], an open source project that uses a bag-of-words approach with different types of 2D image features. In our setup, FindObject2D is configured to use the OpenCV implementation of FAST[15] features on images incoming from the cameras.

Face Recognition is performed with procrob_functional[3], an open source project, which uses the EigenFaces[18] algorithm. The face detection phase is done with the OpenCV implementation of the Viola-Jones[19] approach. Additionally, the output of this module was filtered to avoid false recognition when the person was not perfectly oriented

[1] http://www.github.com/francoisferland/HBBA

[2] https://code.google.com/p/find-object/

[3] https://github.com/procrob/procrob_functional

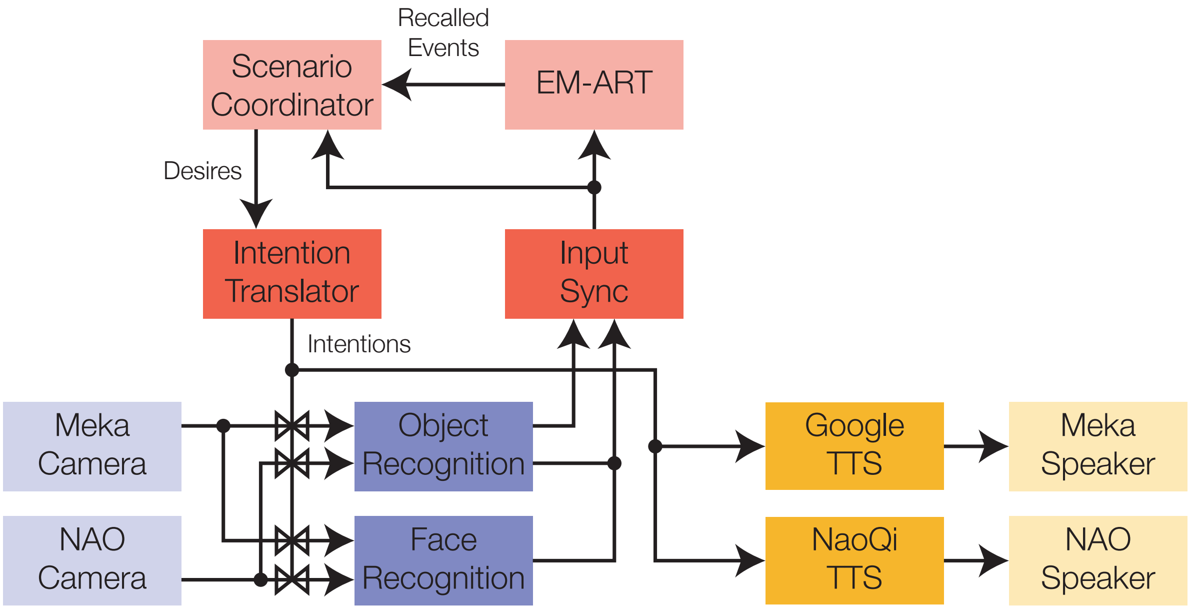


Fig. 2: An instance of HBBA with robot and scenario-specific modules

toward the robot, which occurred mostly for short periods of time when the person was sitting down or standing up, and when the person looked down to locate the object he/she wanted to show to the robot. To remove these false readings from the output stream of either module, the filter kept the last 50 samples in memory and output the most frequent answer in those samples. The samples memory was reset after five seconds of inactivity at the input level so as to avoid influencing the results between two persons.

While Face Recognition worked reasonably well in good lightning conditions, it produced recognition errors in other situations. This occurred more often on the Meka robot than the NAO robot. To avoid this problem, face pictures were added to the Object Recognition database, acting as an alternative solution when the output of Face Recognition proved to be too noisy. In those cases, Face Recognition was completely disabled, and the output of FindObject2D was rerouted to the Face channel of the EM-ART when faces were recognized.

C. Behavior-Producing Modules

In this experiment, a single type of behavior was used to provide Text-To-Speech (TTS) synthesis capabilities to both robots, translating text sentences into audio streams. On the NAO robot, the TTS module provided by the NaoQI framework was used, a single ROS Python node acting as a bridge between both environments.

On the Meka robot, which does not include such a module in its M3 framework, the Google Chrome API over an Internet connection was used instead. Another ROS Python node requested audio samples in MP3 format with this API and played them automatically on the Meka computer.

D. Motivation and Coordination Layer

At the highest level of HBBA, independent Motivation modules create Desires, suggesting the satisfaction or inhibition of Intentions for the robot. Intentions represent the activation of specific modules for behavior production (e.g., going to a specific place and saying phrases) or perception processing (e.g., detecting faces and recognizing speech). In this experiment, only a single Motivation module exists: the Scenario Coordinator. Its role is to produce two static Desires for object and face recognition, and dynamic speech utterances when recognition occurs or events from episodes are recalled. When a person is first detected, the robot is instructed to say ”Hello [person name]”. Then, when objects are recognized, it says ”I recognize the [object name].” Finally, when it anticipates a sequence of events, the robot enumerate the objects associated to it by saying ”I anticipate [object 1], [object 2], [...], and [object N].”

The active Desires set is transformed by the Intention Translator according to a database of strategies and constraints describing the capabilities of the robot currently being used. This database of strategies is how a robot is configured for HBBA. For Object and Face Recognition, this involves selecting strategies that allow data flowing from the proper Sensor modules to the Perception modules by configuring the proper perceptual filter nodes (represented as ⋈ symbols in Fig. 2). Perceptual filter nodes either allow or forbid ROS message from passing between modules. For Behavior modules, activation is done by transmitting sentences generated by the Scenario Coordinator to the proper TTS module.

E. Episodic Memory and Input Synchronization

Before being transmitted to the episodic memory module, shown as EM-ART in Fig. 2, recognized faces and objects are first synchronized by the Input Sync module, which acts as a short-term memory. For each channel, this module keeps the latest recognition data received for up to ten seconds. This helps stabilizing the input to the episodic memory in situations where objects leave the camera frame or occlude the person's face.

The EM-ART model [20], shown in Fig. 3, is made of three layers: Input, Events, and Episodes. In this paper, we use a subset of the implementation presented in [9]. A summary explanation of its algorithm is presented here.

The Input Layer is used to represent the external context information on which to build memories. Information is regrouped in channels in which each node represents the presence (or not) of a known element with an associated activation level. Furthermore, each channel has a relevance value, which enables varying the importance accorded to each channel. In this paper, this layer contains only two channels: faces and objects. Both have the same relevance value, as both types of information were judged to be of the same importance. The output of the Input Layer is the vector x^k and its complement $1 - X^k$ for each channel k, where $x_i^k \in [0, 1]$ is the activation level of channel k for element i, thus encoding both the presence and absence of individual input elements such as a specific person or object.

The Events Layer contains the nodes associated to events experienced, which are derived from the patterns of activated nodes in the Input Layer. In this paper, events represent either the presence of a single person or pairs of one person and one object. Events are automatically recognized or created by trying to match the X^k vectors experienced at the Input Layer with a single node j at the Events Layer. First, nodes are activated by the choice function T_j:

$$T_j = \sum_k \gamma^k \frac{|x^k \wedge w_j^k|}{a^k + |w_j^k|} \tag{1}$$

where $\gamma^k \in [0, 1]$ and $a^k \geq 0$ are the contribution and choice parameters, w_j^k is the weight associating input channel k with event node j, the fuzzy AND operation $\wedge$ is defined by $(p \wedge q)_i \equiv min(p_i, q_i)$ and the norm $|.|$ is defined by $|p| \equiv \sum_i p_i$. The event node with the highest value T_j is then selected, and its resonance is tested with the match function m:

$$m_j^k = \frac{|x^k \wedge w_j^k|}{|x^k|} \geq \rho^k \tag{2}$$

where $\rho_k \in [0, 1]$ the vigilance parameter. Thus, if m_j^k exceeds ρ_k, node j is said to be activated. If it does not match, the next highest value of T_j is tested, and so on until a node can be found. If none of the existing event nodes can be activated, a new node is created.

Each event node has an activation value noted y_j. The latest recognized or created event receives the maximum activation value of $y_j = 1.0$, while the activation values of other events decay according to this relation: $y_j^{(new)} = 0.95 y_j^{(old)}$. Thus, the sequence of events in the Events Layer can be recognized by the growing activation level from early to late events.

The Episodes Layer is made of nodes that categorize the patterns of the activation level of nodes in the Events Layer, defining episodes as temporal sequences of events. While events are automatically created when a new input vector is experienced, episodes are created only when learning is triggered. As the exact length of an episode cannot be known in advance, a mechanism indicating what sequences of events are to be recognized as an episode has to be put in place. In this paper, episodes are separated by empty input vectors. Thus, an episode begins when the person is first recognized and sent to the input layer, and the Input Sync module triggers learning when the person leaves the field of view of the robot, just before sending an empty vector to the Input Layer.

As the robot experiences sequences of events, the same resonance-based matching scheme found in the Events Layer is used to recognize past episodes. This is performed each time the composition of the event activation vector y changes. If resonance cannot be found and the learning trigger is activated, a new episode is created.

Figure 4 illustrates the episode recording process when a person appears in front of the robot and shows it an uninterrupted sequence of three objects. In Fig. 4a, only a single Input Layer node is activated, representing the presence of this person. An event is automatically created for this in the Events Layer, receiving the highest activation value ($y_j = 1$). In Fig. 4b, the person shows a first object to the robot, resulting in the activation of a second node in the objects channel. A new event is also created, and activation levels are updated, lowering the one for the first event. In Fig. 4c and 4d, the person replaces the object, which creates new events and updates the activation levels. At the end of this process, the person leaves the camera frame, which empties the input vector. Learning is automatically triggered by the Input Sync module, and the episode shown in Fig. 3 is recorded.

IV. EXPERIMENT SETUP

To validate if HBBA and its EM-ART module work on both the Meka and the NAO robots, a short scenario was created. Three persons and ten objects were selected and trained for the recognition modules. Figure 5 shows the experimental setup. The objects and the NAO robot were set on a table, with the Meka behind the NAO and the person facing both robots. The relative positions of the robots were set so that the NAO robot was not part of the field of view of the Meka robot.

A. Face and Object Recognition Training

The acquisition of the images for the Face Recognition was done separately in each robot in order to get a better recognition rate. This was necessary as the different resolution and focal length of the cameras would generate different

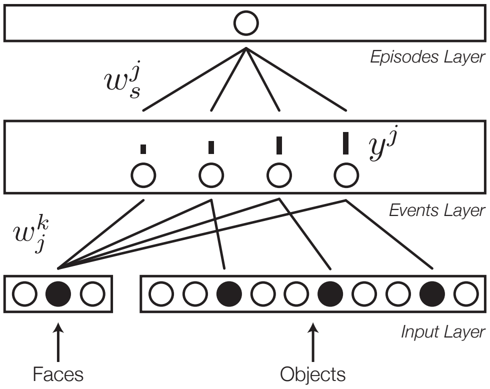


Fig. 3: The EM-ART model with a recalled episode built out of 4 events

features for the recognition algorithm. Three persons were filmed for a time-lapse of 1-2 minutes, varying the angles of the face in front of the robot's camera. These videos were used to perform the training phase, where 50 distinctive images were selected for each person.

Object Recognition Training was achieved by building a data base of 10 objects. These objects came in three categories of sizes: small (e.g., a credit card-sized reward certificate, a candy bar wrapping, a juice box and a 10 cm square booklet), medium (e.g., 4 approximately A4 size books) and large (e.g., two approximately A3 size posters). The focus was on recognizing objects accurately within a large range of distance from the camera. Therefore, multiple snapshots of the objects were taken with both robots at different distances and luminosity conditions. The difference of focal length had less impact with objects than with faces, as all objects could be shown as a flat plane.

B. *Same Robot Validation*

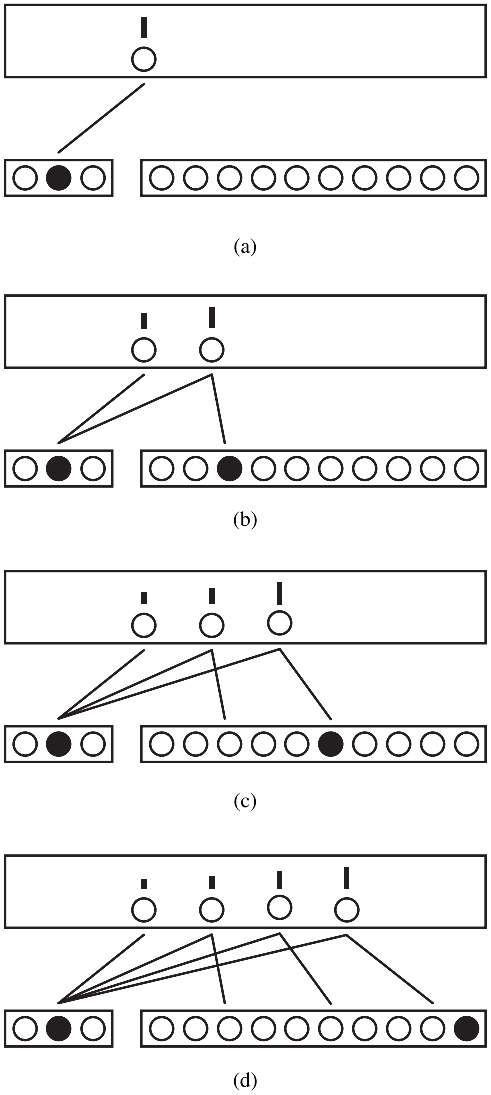
As a first step, episode recording and recalling was tested on one robot at a time. Each person had two sequences of three random objects to show to the robot. The first pass consisted in presenting all six sequences (three participants times two sequences of three objects) once to a single robot, then repeating the same sequences to verify if objects could be anticipated. The complete procedure was repeated on the second robot, and the recalling rate was measured.

C. *Different Robot Validation*

As a second step, another set of six sequences of events was presented to a single robot. However, the procedure was interrupted, the state of the EM-ART staying intact, and the six sequences were presented again but to the second robot. The goal was to validate if one robot could correctly recall episodes experienced with the other robot. The procedure was repeated, inverting the order of robots.

V. Results and Discussion

Table I shows the recall rate of episodes for all the experiments depending on the location where the recording and recalling occurred.


(a)

(b)

(c)

(d)

Fig. 4: An example of the process leading to the recording of an episode

Both 5/6 results can be explained because of a temporary failure of the Face Recognition module. As the person would sometimes become undetected and thus impossible to recognize, it created a point in time where the EM-ART input vector was empty. This triggered episode learning and split the sequence in two separate episodes. The 4/6 (5/5) result occurred when two sequences, both associated with the same person, were merged together by the EM-ART. When it was time to recall objects from this merged episode, five objects were recalled when the person presented the first item of either sequences. This means the model running on the Meka correctly associated the objects to the person who showed them to the NAO, but could not distinguish between the two sequences of three objects. In all successful cases, recalled occurred after the first object was shown, thus making the robot announce its anticipation of the two other objects.

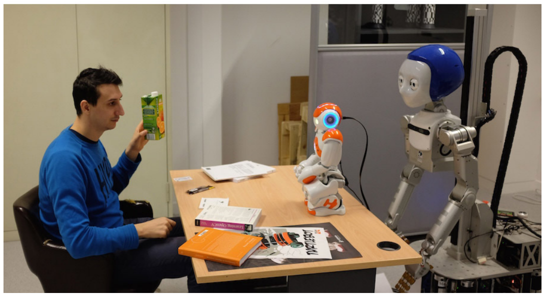


Fig. 5: The experimental setup

TABLE I: Experiment results

	Recorded on NAO	Recorded on Meka
Recalled on NAO	5/6	5/6
Recalled on Meka	4/6 (5/5)	6/6

VI. CONCLUSION AND FUTURE WORK

This paper described the first step of the adaptation of an architecture named HBBA for two robots, the Meka M-1 robot and the NAO robot from Aldebaran, both different from the one that hosted its original development. An experiment scenario showed that the Adaptive Resonance Theory-based episodic memory integrated into HBBA was able to record and recall sequences of events on both robots. Furthermore, sequences of events recorded on one robot have been correctly recalled on the second one, and vice versa.

In future work, we first aim to validate how behaviors within this architecture can be generalized between robots with vastly different actuator structures, and how this affects non-verbal communication. Then, we plan to exploit the anticipation capabilities of the episodic memory to provide natural, adaptive behaviors in long-term human-robot interactions. An important component of this work will include encoding the emotions of the person interacting with the robot (perceived by techniques such as facial action units [3] or prosody variations in speech [21]) as an input channel to the EM-ART. For instance, this could allow the robot to recognize behavioral patterns that lead to negative emotions in the past, and thus avoiding their repetition. We believe that enabling service robots to learn from their past experiences and being able to situate those experiences in an emotional context to be a key in order to provide natural interactions.

ACKNOWLEDGMENT

This work is funded by the Romeo2 project, (http://www.projetromeo.com/), BPIFrance in the framework of the Structuring Projects of Competitiveness Clusters (PSPC). We would also like to thank Flavian-Costin Manea for his help in this project.

REFERENCES

[1] R. A. Brooks, “New approaches to robotics,” *Science*, vol. 253, no. 5025, pp. 1227–1232, 1991.

[2] G. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.

[3] P. Ekman and W. V. Friesen, *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press, 1978.

[4] F. Ferland, D. Létourneau, A. Aumont, J. Frémy, M.-A. Legault, M. Lauria, and F. Michaud, “Natural interaction design of a humanoid robot,” *Journal of Human-Robot Interaction*, vol. 1, no. 2, pp. 118–134, 2012.

[5] N. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.-J. M. Kruijff, M. Brenner, G. Berginc, and D. Skocaj, “Towards an integrated robot with multiple cognitive functions,” in *Proceedings of the 22nd National Conference on Artificial Intelligence*, vol. 7, 2007, pp. 1548–1553.

[6] N. Hawes and J. Wyatt, “Engineering intelligent information-processing systems with cast,” *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 27–39, 2010.

[7] N. Hawes, J. Wyatt, and A. Sloman, “Exploring design space for an integrated intelligent system,” *Knowledge-Based Systems*, vol. 22, no. 7, pp. 509–515, 2009.

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

[9] F. Leconte, F. Ferland, and F. Michaud, “Fusion adaptive resonance theory networks used as episodic memory for an autonomous robot,” in *Artificial General Intelligence*, ser. Lecture Notes in Computer Science, B. Goertzel, L. Orseau, and J. Snaider, Eds. Springer International Publishing, 2014, vol. 8598, pp. 63–72.

[10] M. J. Matarić, “Situated robotics,” *Encyclopedia of cognitive science*, pp. 25–30, 2002.

[11] F. Michaud, C. Côté, D. Létourneau, Y. Brosseau, J.-M. Valin, É. Beaudry, C. Raïevsky, A. Ponchon, P. Moisan, P. Lepage, *et al.*, “Spartacus attending the 2005 AAAI conference,” *Autonomous Robots*, vol. 22, no. 4, pp. 369–383, 2007.

[12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” in *Open Source Software Workshop at the International Conference on Robotics and Automation*, 2009.

[13] A. K. Ramaswamy, B. Monsuez, and A. Tapus, “Solution Space Modeling for Robotic Systems,” *Journal for Software Engineering Robotics*, vol. 5, no. 1, pp. 89–96, 2014.

[14] A. Ramaswamy, B. Monsuez, and A. Tapus, “Saferobots: A model-driven framework for developing robotic systems,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1517–1524.

[15] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[16] S. Taylor, C. Vineyard, M. Healy, T. Caudell, N. Cohen, P. Watson, S. Verzi, J. Morrow, M. Bernard, and H. Eichenbaum, “Memory in silico: Building a neuromimetic episodic cognitive model,” in *Proceedings of the World Congress on Computer Science and Information Engineering*, vol. 5, 2009, pp. 733–737.

[17] E. Tulving, “Precis of elements of episodic memory,” *Behavioral and Brain Sciences*, vol. 7, no. 3, pp. 223–268, 1984.

[18] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[19] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, vol. 4, pp. 34–47, 2001.

[20] W. Wang, B. Subagdja, A. Tan, and J. Starzyk, “Neural modeling of episodic memory: Encoding, retrieval, and forgetting,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1574–1586, 2012.

[21] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, “A survey of affect recognition methods: Audio, visual, and spontaneous expressions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.