4th International Symposium on New Frontiers in Human-Robot Interaction
21–22 April 2015, Canterbury, UK
M. Salem, A. Weiss, P. Baxter, K. Dautenhahn (Eds.)

Embodiment, emotion, and chess: A system description

Christian Becker-Asano and **Nicolas Riesterer** and **Julien Hué** and **Bernhard Nebel**[1]

Abstract. We present a hybrid agent that combines robotic parts with 3D computer graphics to make playing chess against the computer more enjoyable. We built this multimodal autonomous robotic chess opponent under the assumption that the more life-like and physically present an agent is the more personal and potentially more effective the interaction will be. To maximize the life-likeness of the agent, a photo-realistic animation of a virtual agent's face is used to let the agent provide verbal and emotional feedback. For the latter an emotion simulation software module has been integrated to drive the agent's emotional facial expressions in parallel to its verbal utterances.

1 Introduction

Chess has been called the "Drosophila of artificial intelligence" [1] meaning that in the same way as the drosophila melanogaster has become the model organism for biological research, chess served at least for many years as a standard problem for artificial intelligence research. When in 1997 Garry Kasparov, who was ranked first at that time, lost against IBM's supercomputer "Deep Blue" [10], this problem was assumed to be solved and chess engines would nowadays outclass the best players. Altogether this triggered researchers to shift their attention to other games, such as Go. Today, for a casual chess player it can be rather frustrating to play against the computer, because he or she will lose most of the times and the computer moves its pieces with seemingly no hesitation.

Recently it was found, however, that different embodiments of the computer opponent change a human chess player's motivation to engage in a game of computer chess. These attitude changes are rooted in the humans' tendency to treat machines as social actors and this effect seems to be stronger the more human-like the machine is designed to appear [16]. With our development of the hybrid chess-playing agent MARCO, the Multimodal Autonomous Robotic Chess Opponent, we aim to investigate this research question.

The remainder of the paper is structured as follows. After discussing related work in the next section, our general motivation is explained and two research questions are introduced. Then the Elo rating will be explained together with how the employed chess engine evaluates board positions. Subsequently, MARCO's hardware components are detailed, before the interconnection of its software components is laid out. Then, the complete system is explained. Finally, we present our ideas concerning experimental protocols for evaluating MARCO. We conclude this paper with a general discussion.

[1] Artificial Intelligence Lab, University of Freiburg, 79110 Freiburg, Germany, email: [basano,riestern,hue,nebel]@cs.uni-freiburg.de

2 Related work

This section describes research projects involving chess playing robots [15, 18, 13]. They aim to answer different research questions and, therefore, they employ systems of different size and complexity.

"Gambit" is a good example for an engineer's solution to an autonomous chess-playing robotic system [15]. With their "robot manipulator system" the authors created a "moderate in cost" (i.e. 18K USD) manipulator that is able to play chess with arbitrary chess sets on a variety of boards without the need to model the pieces. Although their system does not have any anthropomorphic features, it includes a "natural spoken language interface" to communicate with the human opponent. Most importantly, "Gambit" tracks both the board and the human opponent in real time so that the board does not need to be fixed in front of the robot. With its available six degrees of freedom (DoF) and the USB camera mounted on top of its gripper the robot arm reliably grasps a wide array of different chess pieces, even if they are placed poorly. In result, it outperformed all robotic opponents at the 2010 AAAI Small Scale Manipulation Challenge. Unfortunately, no data on human players' enjoyment is available.

In contrast to the remarkable technical achievements behind the development of "Gambit", the "iCat" from Philips was combined with a DGT chess board to investigate the influence of embodiment on player enjoyment in robotic chess [13]. The authors conducted a small-scale empirical trial with the emotional iCat opponent either presented in its virtual or robotic form. Using a modified version of the GameFlow model [20], it was found that overall the virtual version is less enjoyable than the robotic one. A subsequent long term study [14] with the robotic iCat playing chess repeatedly against five children showed, however, that these children lost interest in the robot. Presumably, iCat's complete lack of any manipulation capability together with its cartoon-like appearance let the children ignore the robot completely after the initial curiosity is satisfied.

Similar to our approach, Sajó et al. [18] present a "hybrid system" called "Turk-2" that consists of a "mechanically simple" robot arm to the right of the human player and a rather simple 2D talking head presented on a computer display. "Turk-2" can analyze three emotional facial expressions, namely *sad*, *neutral*, and *happy*, and additional image processing enables the system to monitor the chess board. Interestingly, the authors decided to artificially prolong the system's "thinking time", details of which are unfortunately not reported. The transitions between the talking head's facial expressions *neutral*, *sad*, *happy*, and *bored* are controlled by a state machine that takes the human's emotion (as derived from its facial expression) and the game state into account. Similar to our approach, the talking head will change into a bored expression after some time without input has passed. An empirical study on the effect of the presence of the talking head revealed that without the talking head the players mostly ignored the robotic arm to the right of them, even when it was moving. With the talking head in front of them, however, the players not only looked at the talking head but also started smiling and laughing.

Regarding the effects of a virtual agent's facial expression of emotions on human performance in a cognitive task, an empirical trial resulted in no significant differences [8]. In addition, the study showed that for such a serious task it made no difference, if the agent's emotions were generated based on a set of hard-coded rules or by making use of a sophisticated and complex emotion simulation architecture. The authors speculate that a less cognitively demanding and more playful task might be better suited to search for such effects.

A prototype of the MARCO system has been demonstrated recently at an international conference [17] and, although conference attendees clearly enjoyed playing and loosing against the agent, several opportunities to improve the system were mentioned. The most noticeable deficiency seemed to be the use of a much too small display for presenting the agent's virtual face. Accordingly, our system now employs a much bigger display.

3 Motivation and research questions

These previous results in combination motivated us to include the following features in MARCO, our Multimodal, Autonomous, Robotic Chess Opponent:

1. A low-cost robotic arm that enables MARCO to autonomously move the chess pieces instead of having to rely on the human opponent's assistance (as in [13])
2. A custom built, robotic display presenting a highly anthropomorphic virtual agent's head to realize a hybrid embodiment combining the best of both worlds, cp. [13, 18]
3. A flexible software architecture that relies on an established emotion simulation architecture as one of its core modules (following up on [8])

The resulting MARCO system will help answering research questions that are motivated by the previous work presented above:

1. Is it more enjoyable to play chess against the robotic arm with or without the virtual agent?
2. Is it more enjoyable to play against the hybrid agent (i.e. the robotic arm with the virtual agent) when the agent expresses emotions as compared to when it remains equally active but emotionally neutral?
3. Is the most human-like and emotional agent evaluated as more social/mindful than the less complex/human-like versions of it? Does this subjective evaluation depend on how experience the human chess player is?

The first question will provide a baseline for the hardware components of our system and will be compared with those reported in [18] with regard to "Turk-2". It is not taken for granted that a more complex system will always be preferable to a simpler system from the perspective of a human player. The second question, however, is targeting the role that artificial emotions might or might not play and it is motivated by previous results [8]. Finally, MARCO allows us to tackle systematically the general question of how and when "mindfullness" is ascribed to machines [16].

4 Background and Preliminaries

4.1 Elo rating

The skill of chess players is usually measured in terms of a single integer value, the so-called Elo Rating [12]. It represents the relative strength of a player, the higher the better, and it increases or decreases with his or her chess match results. Currently, ELO rating in chess goes from 1000 (complete beginner) to 2880 (Magnus Carlsen World Champion).

Differences in the evaluations of our system might correlate with or even depend on the ELO ratings of the human players. In addition, our system might be used as a virtual coach for novice players to improve their chess skills and the ELO rating provides a standard means to compare player strength before and after training.

4.2 Chess Engine

Computer chess engines evaluate the board position using an alpha-beta algorithm with a depth d given as parameter based on a number of criteria like: pieces left on the board, activity of these pieces, security of the king, etc. The greater the depth the more precise is the evaluation. The position evaluation function results in a real number e ranging from $[-\infty, +\infty]$ where 0 means that the position is equal, $-\infty$ that black is winning and $+\infty$ that white is winning. A $+1$ valuation roughly represents the advantage equivalent to a pawn, $+3$ to a knight or a bishop, and so on according to the standard valuation of chess pieces.

We denote by $e_{t,d}$ the evaluation given by the chess engine at move t with depth d. We write e when it is clear from the context. In practice, once $|e| \geq 5$ the game is more or less decided.

Our first prototype [17] was based on the TSCP chess engine [2] for its simplicity and in order to make our results comparable to previous work on the iCat playing chess [13], for which the same engine was used. The communication between the user and the TSCP chess engine is handled by the XBoard Chess Engine Communication Protocol [3]. Originally implemented as a means to facilitate communication between the GNU XBoard Chess GUI and underlying chess engines, this plain text protocol allows for easy information exchange in a human readable form.

Our modular software architecture allows us, however, to plug in other chess engines. The more advanced Stockfish chess engine [4], for example, would allow us to adjust the strength of MARCO's play dynamically.

5 Hardware components

The complete setup is presented in Figure 1. The hardware used comprises a custom designed, 15.6 inch pan-tilt-roll display to present the virtual agent's face, a robotic arm to the right of the agent to move the chess pieces, and a digital chess board (DGT USB Rosewood) with a chess clock. Each of these components will be described next.

5.1 The pan-tilt-roll agent display

The pan-tilt-roll display component features a 15.6 inch upright TFT LCD display with a physical resolution of 1920×1080 pixels and 18bit color depth, cp. Fig. 1. It is positioned opposite of the human player to give the impression of the virtual agent overlooking the complete chess board. Three Dynamixel AX-12A servos (cp. Fig. 2(a)) are connected to USB2Dynamixel interface to allow for control over the display's orientation during the game along all three axes. Thereby, for example, the agent can follow its own arm's movements dynamically as presented in Fig. 2(b).

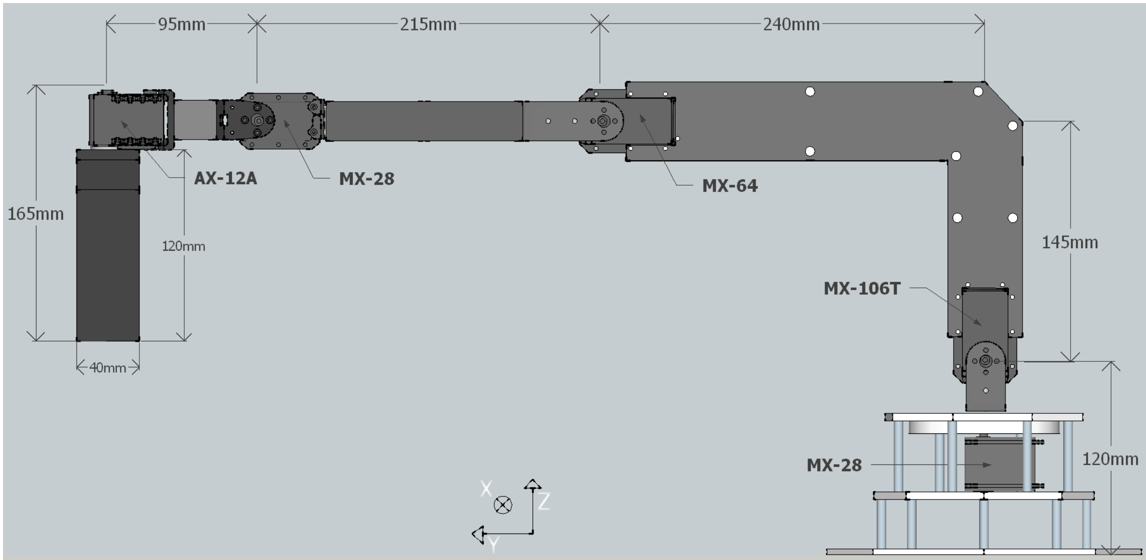
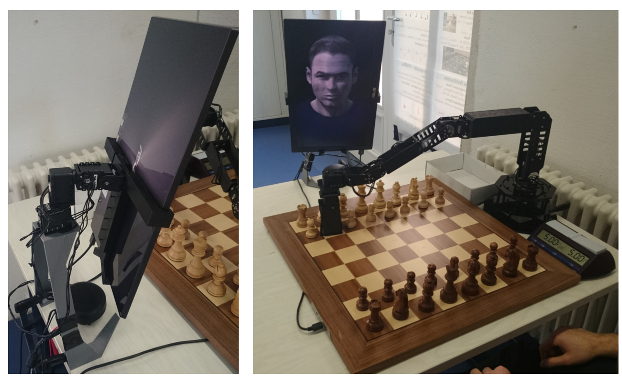


Figure 3. A schematic of the robotic arm with annotations of link lengths and Dynamixel servos used for each joint position

Figure 1. The pan-tilt-roll agent display, the robotic arm, and the digital chess board together with the digital chess clock

5.2 The robotic arm

The hybrid agent's robotic arm is a modification of the "WidowX Robotic Arm Kit Mark II" [5] available from Trossen Robotics. Apart from the rotational base all other parts needed to be extended to allow the agent to pick-and-place all pieces on any of the 64 squares of the board. The upper arm was extended to measure $240mm$, the



(a) Detail view

(b) Front view

Figure 2. Pan-tilt-roll mount of the 15.6 inch display presenting the virtual agent's face

forearm to measure $215mm$ and the gripper needed to be prolonged to $120mm$ (cp. Fig. 3). These extensions for the arm as well as the extra parts to realize the display mount were printed with a MakerBot 3D printer. Five Dynamixel servos move the robot's arm, cp. Fig. 3. For the base and wrist two MX-28 servos are used. An MX-64 servo moves the robot's elbow and an MX-106 servo its shoulder. The modified gripper is opened and closed by an AX-12A servo, cp. Fig. 4. It can reliably pick-and-place all Staunton chess pieces on the DGT board regardless of their height or size.

5.3 The DGT digital chess board

The DGT chess board is a wooden board with standard Staunton pieces and $55mm \times 55mm$ squares. Each piece is equipped with a unique RFID chip that makes it recognizable. The board is con-

(a) Open state (b) Closed state

Figure 4. The two states of the robot's custom designed gripper picking up a white bishop

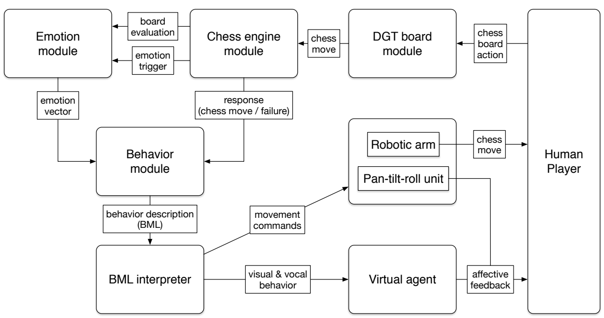


Figure 5. An outline of the software modules and their connections

nected to the computer with a USB cable, and it transmits the position in FEN format to the DGT board module every time a change is performed.

6 Software components

Except for the external MARC framework (see Section 6.3), all components are implemented in C++ using Qt5 [7] in combination with the Robot Operating System (ROS; [6]) to achieve a modular design and cross-platform functionality. The hardware components (i.e. the DGT chess board and the Dynamixel servos) are encapsulated into ROS nodes to establish a flexible communication infrastructure.

6.1 Overview of system components

The following five main software components can be distinguished, which are connected by the ROS message protocol (cp. Fig. 5):

- A DGT board module to detect moving pieces on the physical chess board
- A Chess engine model for position evaluation and chess move calculation
- An Emotion module to simulate MARCO's emotions
- A Behavior module to integrate the chess move with emotional states into a behavior description
- A Behavior Markup Language (BML) Interpreter to prepare the multimodal realization of the behavior
- Robotic components to move the chess pieces on the board and control the virtual agent's pan-tilt-roll unit
- The MARC framework to create the agent's visual appearance on the display

When the human player (cp. Fig. 5, right) performs her move, the DGT board module recognizes the change on the board, derives the move information by comparing the current board configuration with the previous one, and sends this information to the chess engine module. Here, the chess move is verified for correctness and either (1) a failure state, or (2) the chess engine's move is transmitted as MARCO's response to the behavior model. The board evaluation function of the chess engine also provides the emotion module with input. After the emotion module integrated the board evaluation into the agent's emotion dynamics (see Section 6.2), it concurrently updates the behavior module with a vector of emotion intensities. The behavior module integrates the emotional state information with the move calculation into a behavior description in BML [21]. This description is then interpreted by the BML interpreter to drive the virtual agent's visual and vocal behavior as well as the robotic component's actions. While the robotic arm starts to execute the agent's chess move, the pan-tilt-roll unit moves the display to realize affective feedback in combination with the virtual agent's facial expressions.

6.2 Deriving emotional states

The emotion module (cp. Fig. 5) comprises the WASABI Affect Simulation Architecture [9] to simulate the agent's dynamically changing emotional states. As input WASABI needs *valenced impulses* and expectation-based emotions (e.g., *surprised* and *hope*) need to be *triggered* before they can gain positive intensity.

6.2.1 Emotion dynamics

WASABI is based on the idea that emotion and mood are tightly coupled. The term "mood" refers to a relatively stable background state, which is influenced by emotion arousing events, but changes much more slowly as compared to any emotional state. An "emotion", in contrast, is a short-lived psychological phenomenon that more directly impacts behavior than a mood does.

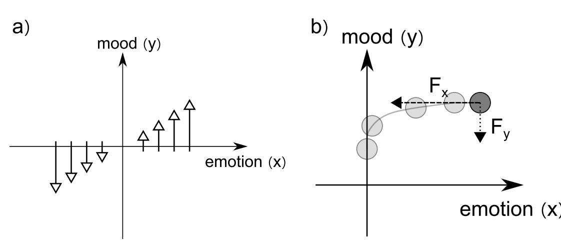


Figure 6. The emotion dynamics of WASABI with (a) the influence of emotional valence on mood, and (b) the effect of the two independent mass-spring systems on the development of the agent's emotional state over time (indicated by the half-transparent circles)

Taking these differences and commonalities as cue, WASABI simulates the positive and negative effects that emotional valence has on mood, cf. Fig. 6a. In addition, mood and emotion are driven back to zero by two forces independently exerted from two mass-spring systems. Notably, the respective spring constants are set such that the resultant force F_x is always greater that the resultant force F_y, because emotions are longer lasting than mood, cp. Fig. 6b.

MARCO's emotional state as represented in Fig. 6b by the circles is updated with 50Hz letting it move through the space over time. The x and y values are incessantly mapped into PAD space to allow for categorization in terms of emotion labels (cp. Fig. 7; see also [9]).

This dynamic process is started by the arrival of a *valenced impulse* from outside of WASABI that instantaneously changes the emotion value (x) either in the positive or negative direction. How these impulses are derived from the progression of the game is described next.

6.2.2 Valenced impulses

The chess engine module continuously calculates board evaluations e_t (at times t during the game). These are converted into *valenced impulses* $val(e_t)$ according to Equation 1.

$$val(e_t) = k \times tanh\left(\frac{e_t}{r}\right) \tag{1}$$

Here, k is a scaling factor and by increasing the denominator $r \in [1, \infty]$ the skewness of the hyperbolic tangent is reduced until a quasi-linear mapping ($val(e_t) = k \times e_t$) is achieved. The hyperbolic tangent is introduced to let us emphasize small values of e_t relative to bigger values of e_t.

For example, choosing $k = 50$ and $r = 2$:

$$\begin{gathered} val(e_t) = 50 \times tanh\left(\frac{e_t}{2}\right) \in (2.5, 25], \\ \forall e_t \in \{x \in \mathbb{R} \mid 0.1 \le x < 1.1\} \end{gathered} \tag{2}$$

Thus, with these constants any value of e_t between 0.1 and 1.1 results in a weak to medium valenced impulse. Observe that $|val(e_t)| \cong 50, \forall e_t \in \{x \in \mathbb{R} \mid |x| > 5\}$, meaning that a winning (or loosing) board configuration results in the maximum impulse of 50 (or minimum impulse of -50, respectively).

Depending on who plays white, the sign of the scaling factor k is adjusted as to map favorable board positions for MARCO to positively valenced impulses and vice versa. That is, if MARCO plays white k is positive, otherwise it is negative. For the time being, MARCO always plays white letting it perform the first half-move.

Inside the emotion module the *valenced impulses* drive the concurrent simulation of the agent's emotion dynamics. In summary, a positive (negative) impulse has the short term effect of increasing (decreasing) the agent's *emotional valence*, which in turn influences the agent's *mood* in the same direction as a long term effect. A simple mathematical transformation into *pleasure* ($P = \frac{x+y}{2}$) and *arousal* ($A = |x|$) values is performed and the emotion module then uses the PAD space (cf. Fig.7) to categorize the agent's emotional state in terms discrete emotions and their intensities. The *dominance* value is changed in accordance with whether it is MARCO's turn ($D = 1$) or not ($D = 0$). Finally, the resulting set of emotions with positive intensities are transmitted to the behavior module.

6.2.3 Mapping onto discrete emotions

In its default configuration, WASABI simulates the primary emotions *annoyed*, *angry*, *bored*, *concentrated*, *depressed*, *fearful*, *happy*, *sad*, and *surprised* as well as the secondary emotions *relief*, *fears-confirmed*, and *hope*; cp. Fig. 7. Five of these 12 emotions (*fearful*, *surprised*, *relief*, *fears-confirmed*, and *hope*) rely on an agent's ability to build expectations about future events, i.e., they are so-called *prospect-based emotions*. For example, one is only surprised about an event, if it is contrary to one's previous expectations, or one fears future events, only if one has reason to expect that bad event is about to happen [9]. Accordingly, in WASABI each of these emotions is configured with zero *base intensity* and needs to be *triggered* (cp. "emotion trigger" in Fig. 5) to give them a chance to gain positive intensity.

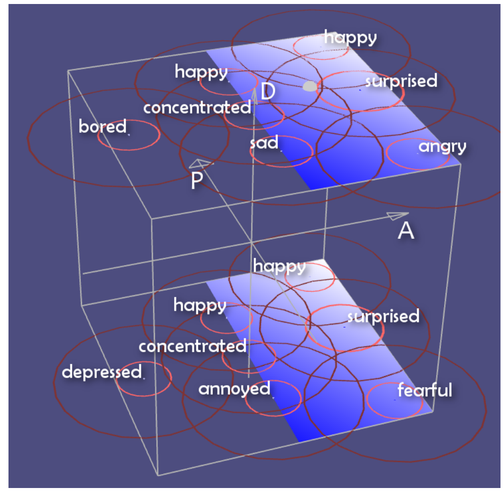


Figure 7. The PAD-space of primary and secondary emotions in WASABI. The primary emotions are distributed as to cover all areas of PAD space. For each of them an activation threshold (outer ring) and a saturation threshold (inner ring) is defined. The two shaded areas represent the distribution of the secondary emotion *hope* in the dominant and submissive subspace, after it was triggered. The grey half-sphere represents MARCO's dynamically changing emotional state. Thus, in this example MARCO would be mildly *happy*, a bit *concentrated*, and quite *hopeful*. If surprise were triggered as well in this moment, MARCO would also be *surprised* to a certain extend.

With respect to chess, our system evaluates the available moves for its opponent. MARCO is able to realize, whenever its last move was less good than previously evaluated, because at time t the evaluation reaches one level deeper into the search tree than at time $t-1$. Accordingly, MARCO might start to fear that the human opponent realizes her opportunity as well. If the evaluation of the situation after the opponent's move is stable, then MARCO's fears are confirmed: the opponent made the right move. On the other hand, if the evaluation comes back to what it was before, i.e., before MARCO made its last move, then the opponent missed the opportunity and MARCO is relieved. The evaluation can be in between these two values and in that case, the agent is neither relieved nor sees its fears confirmed. Nevertheless, the emotion module still receives the negative *valenced impulse* derived from the drop. Formally, Table 1 provides details on how the changing evaluations trigger prospect-based emotions in WASABI.

Notably, the value e_t represents the future directed evaluation of the situation from the robot's perspective. For example, the formula $e_{t-1} - e_t > \epsilon$ lets the behavior trigger *fear* whenever a significant drop in the evaluation function appeared from the previous move to

trigger	*if..*
fear	$e_{t-1} - e_t > \epsilon$
surprise	$\lvert e_{t-1} - e_t \rvert > \epsilon$
fears-confirmed	$fear_{t-1} \wedge (e_{t-1} - e_t < \epsilon)$
hope	$e_{t,d} - e_{t,d-2} > \epsilon$
relief	$fear_{t-1} \wedge (e_t - e_{t-2} < \epsilon)$

Table 1. The conditions under which the prospect-based emotions are triggered in WASABI based on the changes of evaluations over time with ϵ and depth d as custom parameters

Figure 8. The virtual agent expressing *anger*, *neutral*, and *joy* (left to right)

the current one. That is, MARCO realizes at time t that the future seems much worse than evaluated before (in time $t-1$). If subsequently, after the next half-move in $t+1$, the value e_{t-1} turns out to have been correct in the light of the new value e_t (or the situation got even worse than expected), then *fears-confirmed* will be triggered. On the contrary, if it turned out to be much better than expected, *relief* will be triggered. *Surprise* is always triggered when the evaluation changes significantly from one half-move to the next. Finally, *hope* is triggered whenever not taking the full depth of the search tree into account would mean that the key move in the position is hard to reach (requires a computation at depth at least d).[2]

6.2.4 *The emotion vector as input for the behavior module*

It is important to note that, in addition to an emotion being triggered, the *pleasure*, *arousal*, and *dominance* (PAD) values driven by the emotion dynamics must be close enough to that emotion for it to become a member of the *emotion vector* with positive intensity, cp. Fig. 7. Thus, although *surprise* will always be triggered together with *fear*, they will not always both be present in the *emotion vector*, because they occupy different regions in PAD space.

From the *emotion vector* the emotion with the highest intensity is compiled into the BML description driving the MARC framework. The agent comments on particular events like, for example, complimenting the player after it lost a game or stating that the position is now simplified after exchanging the queen.

6.3 The virtual agent provided by the MARC framework

The MARC framework [11] is used to animate the virtual agent, which is presented on the 15.6 inch pan-tilt-roll display facing the human player. The emotional facial expressions (see Fig. 8 for examples) that are provided as part of the BML description are combined inside the MARC framework to create lip-sync animations of emotional verbal utterances. Thanks to the integration of the open-source text-to-speech synthesis OpenMARY [19] the agent's emotion also influences the agent's auditory speech.

[2] An evaluation function is usually set up to an even number, thus the last level of the search tree equals the last two half-moves.

7 Conclusions and future work

This paper detailed the software and hardware components behind MARCO, a chess playing hybrid agent equipped with a robotic arm and a screen displaying a virtual agent capable of emotional facial expressions. A first prototype of the system was demonstrated at an international conference [17] and the experiences gained let to improvements both on concerning the hard- and software components.

Although a limited set of concrete agent behaviors has proven to be fun for the conference participants, we still need to design many more of them. For example, we need to decide which kind of comments are to be given with which timing during the game and how virtual gaze and robotic head movements are to be combined to give the impression of a believable, hybrid agent.

In order to answer the initially stated two research questions, we plan to conduct a series of empirical studies. At first, one group of participants will play against MARCO with the pan-tilt display turned off. Nevertheless, the invisible agent's comments will remain audible in this condition. In the second condition, another group of participants will play against MARCO with an unemotional agent presented on the robotic display. For the third condition, a group of participants will play against the WASABI-driven agent. In all three conditions, player enjoyment will be assessed using the GameFlow [20] questionnaire and video recordings of the human players will be analyzed inspired by [18]. We expect to find significant differences between conditions with the most complete setup (condition three) being most fun for the players.

Nass and Moon claim that imperfect technologies mimicking human characteristics might even increase "the saliency of the computer's 'nonhumanness'." [16, p. 97] In line with their ideas and in addition to the approach outlined above, we plan to compare human-human interaction with human-agent interaction when competing in chess to measure and incessantly improve MARCO's level of human-likeness. This will help to understand how human behavior might be split into computationally tractable components and then realized in robotic agents to improve human-computer interaction.

REFERENCES

[1] `http://aitopics.org/topic/chess`.
[2] `http://www.tckerrigan.com/Chess/TSCP`.
[3] `http://www.gnu.org/software/xboard/engine-intf.html`.
[4] `https://stockfishchess.org/`.
[5] `http://www.trossenrobotics.com/widowxrobotarm`.
[6] `http://www.ros.org/`.
[7] Qt: Cross-platform application and UI framework. http://qt-project.org/, May 2014.
[8] C. Becker-Asano, P. Stahl, M. Ragni, J.-C. Martin, M. Courgeon, and B. Nebel. An affective virtual agent providing embodied feedback in the paired associate task: system design and evaluation. In *Proc. of the 13th. Intl. Conf. on Intelligent Virtual Agents (IVA 2013)*, pages 406–415, Edinburgh, UK, August 2013.
[9] C. Becker-Asano and I. Wachsmuth. Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems*, 20(1):32–49, January 2010.
[10] M. Campbell, A. H. Jr., and F. hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1 – 2):57 – 83, 2002.

[11] M. Courgeon, J.-C. Martin, and C. Jacquemin. MARC: a Multimodal Affective and Reactive Character. In *Proc. 1st Workshop on AFFective Interaction in Natural Environements*, 2008.

[12] A. E. Elo. *The rating of chessplayers, past and present*. Arco Pub., New York, 1978.

[13] I. Leite, C. Martinho, A. Pereira, and A. Paiva. icat: an affective game buddy based on anticipatory mechanisms. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, AAMAS '08, pages 1229–1232, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[14] I. Leite, C. Martinho, A. Pereira, and A. Paiva. As time goes by: Long-term evaluation of social presence in robotic companions. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 669–674, Sept 2009.

[15] C. Matuszek, B. Mayton, R. Aimi, M. Deisenroth, L. Bo, R. Chu, M. Kung, L. LeGrand, J. Smith, and D. Fox. Gambit: An autonomous chess-playing robotic system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4291–4297, May 2011.

[16] C. Nass and Y. Moon. Machines and mindlessness: Social responses to computers. *Journal of Social Issues*, 56(1):81–103, 2000.

[17] N. Riesterer, C. Becker Asano, J. Hué, C. Dornhege, and B. Nebel. The hybrid agent MARCO. In *Proc. of the 16th Intl. Conf. on Multimodal Interaction*, pages 80–81. ACM, 2014.

[18] L. Sajó, Z. Ruttkay, and A. Fazekas. Turk-2, a multi-modal chess player. *International Journal of Human-Computer Studies*, 69(7–8):483–495, 2011.

[19] M. Schröder. OpenMARY sources. https://github.com/marytts/marytts, April 2013.

[20] P. Sweetser and P. Wyeth. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, 3(3):3–3, 2005.

[21] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, Z. Ruttkay, K. Thóisson, H. Welbergen, and R. J. Werf. The behavior markup language: Recent developments and challenges. In *Intelligent Virtual Agents*, volume 4722 of *LNCS*, pages 99–111. Springer Berlin Heidelberg, 2007.