Towards Empathetic Care Robots

Stefania Costantini[1], Pierangelo Dell'Acqua[2], Abeer Dyoub[1] and Andrea Monaldini[1]

[1]*Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy*
[2]*Dept. of Science and Technology, Linköping University, Sweden*

Abstract
Care Robots are the future of healthcare giving for patients with chronic diseases and disabilities. For improving the quality of care-giving and the trustworthiness of those robots, they should be equipped with emotion recognition capabilities and empathetic behavior. In this work, we propose an framework for an empathy module to be incorporated in every care robot, and then demonstrate the effectiveness of our proposal by means of an example.

Keywords
Human-centered computing, human-automation interaction, affective computing, behavior trees

1. Introduction

Healthcare is shifting into patient-centered healthcare with the objective to empower patients to become active participants in their care and to ensure better health outcomes. However, the nonclinical needs of patients mental health and well-being are frequently overlooked by contemporary patient-centered healthcare models.

The COVID19 pandemic has accelerated the development of robots and virtual assisted living that can help care for persons with disabilities and aging adults both physically and emotionally. Given the intimate human-machine interaction in the case of care robots, it has become fundamental for these robots to demonstrate an empathetic behavior. This would result in more productive and delightful interaction that contribute to the patient well-being and mental health and to the trust relation between the patient and the machine.

Artificial Empathy (AE) refer to the development of AI systems, such as care robots or virtual agents, that are able to detect and respond to human emotions in an empathetic way. Interest in empathetic robots is growing in academia and industry in the last years.

The patient and her/his care robot (CR) can b seen as two agents in a Multi-agent system (MAS). To achieve better results via cooperation and enhance the mutual trust, agents interacting with other agents and in particular with humans must be able to reason about what these other agents should and can do, because the robot should support the human to accomplish her tasks. Several agent-oriented programming languages and systems exist, many of them based upon computational logic (cf., e.g., [1] for recent survey on such languages). We concentrate on such approaches, as logic-based agents are in principle verifiable and thus trustworthy, and more capable that in other approaches to provide the user with explanations, as logical inference can be transposed into natural language (see, e.g., [2], Chapter VII).

Theory of Mind (ToM) is starting to be applied to robotics [3]. It is often linked to the so-called Affective Computing, which is a set of techniques able to elicit a human's emotional condition from physical signs, to enable the system to respond intelligently to human emotional feedback, and thus to enhance ToM activities by providing it with perceptions related to the user's emotional signs. In order to render these robots acceptable and even appreciated by users, they will have to be programmed so as to mimic basic social skills and behave in a socially acceptable manner. This means that their behaviour should be to some extent predictable by the user and conformant to social and ethical standards [4].

Virtual Reality (VR) defines as a technology that creates simulated environments to mimic real-world situations.

Since the use of VR can turn threatening and tedious conditions into safe and enjoyable states; in recent years, employing this technology has been considered for the treatment of many mental illnesses especially for anxiety disorders [5]. One approach that can be implemented in VR to treat anxiety is exposure therapy (VRET), it is client-centered and helps clients confront fear-inducing stimuli through guided exposures and is often paired with cognitive–behavioral therapy.

For the sake of improving care giving by care robots, in this work, we propose a framework for an empathy management module, based upon an enhanced notion of *Behavior Trees.*

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
***Corresponding author.
stefania.costantini@univaq.it (S. Costantini);
pierangelo.dellacqua@liu.se (P. Dell'Acqua);
abeer.dyoub@univaq.it (A. Dyoub);
andrea.monaldini@student.univaq.it (A. Monaldini)

2. Background: Behavior Trees

Behavior Trees (BTs) were invented as a tool to enable modular AI in computer games. A behavior tree is essentially a mathematical model of plan execution, where each element (task, action, etc.) of a plan is associated to a node in the tree. Their strength comes from their ability to create complex tasks composed of simple tasks without worrying how the simple tasks are implemented. In the last decade BTs received an increasing amount of attention both in computer science, robotics, control systems and video games. For comprehensive survey of BTs in Artificial Intelligence and Robotic applications see [6].

In this section, we introduce a definition of BTs based on the description of Champandard and Knafla [7, 8]..

A BT is a directed acyclic graph consisting of different types of nodes, each one associated with executable code (where such code enacts an element composing a plan). In most cases, a BT is tree-shaped, hence the name. However, unlike a traditional tree, a node in a BT can have multiple parents which allows the reuse of that part of the BT. The traversal of a behavior tree starts at the top node. When a node is traversed the associated code is executed (we say for short that the node is executed), returning one of the three states: success, failure or running. In our case, a BT is composed of the following types of nodes, where the type denotes the kind of task related to node execution:

2.1. Leaf Nodes

Action: An action represents a behavior that the character can perform. The action returns success, failure, or running state. An action is depicted as a white, rounded rectangle.

Condition: A condition checks an internal or external state. It returns either success or failure. A condition is represented as a gray, rounded rectangle.

2.2. Inner Nodes

Sequence Selector: A sequence selector is a node that typically has several child nodes that are executed sequentially. If every child node returns success, then this selector returns success. Should any child fail, the selector immediately returns failure. If a child returns running, the selector also returns running. A sequence selector is depicted as a gray square with an arrow across the links to its child nodes.

Priority Selector: A priority selector has a list of child nodes which it tries to execute one at a time, with respect to the specified order, until one of them returns success. If none of the children succeeds, then the this selector returns failure. If a child is running, it returns running.

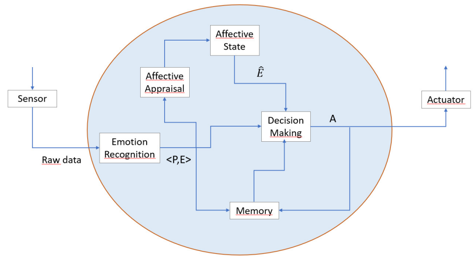

Figure 1: A general architecture for emotional empathetic agents.

A priority selector is represented with a gray circle with a question mark in it.

Parallel Node: A parallel node executes all of its child nodes in parallel. A parallel node can stop executing its child nodes. One may specify the number of child nodes that must execute successfully for the parallel node to succeed, and those that must fail in order for the parallel node to fail. A parallel node is depicted as a gray circle with a P in it.

Decorator: A decorator is a node that acts as a filter that places certain constraints on the execution of its single child node without affecting the child node itself. Decorators are represented as diamonds with descriptive text inside.

3. A framework for Empathetic Agents

In this section, we present the proposed framework for emotional empathetic agents, depicted in Figure 1 and discuss its main components. *Sensors* An agent perceives its environment through sensors and acts upon it through actuators.

Emotion Recognition: this module receives the raw data from the sensory input and processes it to synthesize the user's affective state E. The output $< P, E >$ differentiates between the input data P from the sensor about the environment and the synthesized emotional state E.

Affective Appraisal refers to the process in which events from the environment are evaluated in terms of their emotional significance. Appraisal theory is the theory in psychology that emotions are extracted from our evaluations (appraisals or estimates) of events that cause specific reactions in different people.

Affective State The term affective state refers to how an entity is currently feeling, that is the product of its emotions at a certain moment in time. Within an emotional agent architecture [9], emotions were represented as signals[1] coming from the Affective Appraisal module.

[1]The signals correspond to what in neuroscience is the concentration

The set of all signals of the same type forms the corresponding emotional state. Each signal has the form of a sigmoid curve and consists of the following phases: delay, attack, sustain and decay.

Decision Making This module is responsible for selecting the next action to execute. It receives inputs from the Emotion Recognition module as well as from the Affective State and the Agent Memory module. Note that here with E we indicate the emotional state of the human that interact with the automation (namely, the agent), and with $\hat{E}$ we indicate the emotional state of the agent (that is, the automation).

Agent Memory The knowledge base represents the memory of the agent. Here, all information from sensory inputs p from the environment and user's emotional state E as well the actions A selected to be executed are stored with a time stamp.

Actuators The actions A selected by the Decision Making module are passed to actuators, whose role is to execute them on the environment. Actions come together with an emotion encoding to display agents emotions via verbal or visual communication. Each actuator, depending on the type of application, must be equipped with the ability to render the emotional aspect of actions. One example could be a verbal communication to a human while having a smiling face.

4. Emotional Behavior Trees

It is not straightforward to couple behavior trees with emotions to mimic human emotional decision making. If we wish to have a natural and interesting agent's behavior, it is important that characters behave in an emotional way. It could be claimed that it is possible to incorporate emotions into behavior trees by merely using emotions in the conditions. However, doing so may create large cumbersome behavior trees that are difficult to manage. For each behavior, a specific set of conditions would have to be placed on emotional states. These conditions would most likely take the form of checking the emotional values against a fixed threshold, which would disable a subtle emotional effect on decision making. Thus, this approach would most likely lead to a large behavior tree with numerous nested conditions, making it difficult to construct and manage. Furthermore, in this work we focus on emotion-based interaction between humans and machines, and human (the end user), with emotions modeled as condition-action, would certainly feel that the system is programmed to react to her/his inputs not in a genuine emotional way, but rather in a rational way as a machine typically does.

---

of certain chemical substances in human brain. The signal we use is a simplified representation of the concentration levels.

4.1. Emotional Selector

To take emotions into consideration, Johansson and Dell'Acqua [10] extended the definition of behavior trees and introduced a new type of selector, called the *emotional selector*. They called the resulting model the *emotional behavior tree* (EmoBT).

Emotional Selector: reorders its child nodes according to a number of identified relevant factors and the affective state of the agent (see section 5). Once the ordering has been established based upon the probabilities of nodes, the emotional selector behaves as a priority selector. When it completes its execution, and re-execute, the ordering of the nodes must be re-calculated. An emotional selector is represented with a gray circle with the character 'E' in it.

5. Modelling VRET-Companion Behavior via EmoBTs

We present here a simple VRET-Companion behavior scenario with an emotional behavior tree. This example aims to illustrate the usefulness of our model in a VRET scenario. VERT-Companion is a virtual character that play the role of user companion in a virtual reality settings. This character try to approach the user (represented as another character) and interact with her/him.

Similar to [10], we will consider three relevant aspects for our application; Risk, Time and Planning. Below we show how to incorporate these three aspects into EmoBTs by following methodological steps.

1. *Objectives*: To model VRET-Companion characters.
2. *Relevant aspects* $R = \{Risk, Time, Planning\}$.
   Risk perception: The perceived risk of an action is greatly influenced by emotions [11]. Studies by Lerner and Keltner [12] have shown that happy and angry people are willing to accept greater risks, while fearful people are more pessimistic. Maner et al. [13] shows that anxiety is connected to risk-avoidance, while sadness allows for greater risks, but instead gives a focus on high rewards.
3. $E = \{fear, fatigue, sadness\}$. Only three emotions were identified for simplicitly of exposition.
4. Definition of $Risk$-value (Risk Assessment): Risk has to do with how dangerous the character believes a situation is. A risk value is between 0 and 1; 0 being no risk at all, and 1 being extremely dangerous. The risk value measures the probability of risk. EmoBTs cannot reason about the risk of performing an action, but we allow the designer to add a risk value to each leaf

node in the tree, and derive the associated risk for the inner nodes.

Action: An action has a risk value that is set by the designer (0 by default).

Condition: A condition has a risk value that is set by the designer (0 by default).

Sequence Selector: Since a sequence selector performs every child node of the sequence, the risks of every child nodes must be combined. The overall risk value is calculated as:

$$Risk_j = 1 - \prod_{i=1}^{N} (1 - Risk_i)$$

where N is the number of child nodes of j.

Priority Selector: A priority selector j only executes one of its child nodes. Since we cannot determine in advance which node will be executed, we define the risk value of j as the average of the risk of every child node i:

$$Risk_j = \frac{\sum_{i=1}^{N}(1 - Risk_i)}{N}$$

Parallel Node: Since all of the child nodes of a parallel node j are executed, the risk is defined as:

$$Risk_j = 1 - \prod_{i=1}^{N} (1 - Risk_i)$$

where N is the number of child nodes of j.

Decorator: The risk value of a decorator is the same as the one of its child node.

5. To mimic how affective states influence decision making, we introduce emotional weights for every relevant factor. Below we show the Risk factor. Let $e_1^+, \dots, e_M^+$ (resp. $e_1^-, \dots, e_N^-$) be the values of the emotions that positively (resp., negatively) affect the perception of risk. We define the emotional weight for risk as:

$$E_{Risk} = \frac{\sum_{i=1}^{M} e_i^+}{M} - \frac{\sum_{j=1}^{N} e_j^-}{N}$$

6. For every aspect $r \in R$ we define the weight $W_{r,i}$ of every child node i of any emotional selector. We consider the $Risk$ aspect. The weight for risk for a child node i is calculated as:

$$W_{Risk,i} = (1 - E_{Risk} \times \delta) \times Risk_i$$

where $Risk_i$ is the risk value for the child node i. Note that $W_{Risk,i}$ should be clamped to the interval [0; 1] since it represents a probability. The variable δ determines how much emotions affect the weights. Its value must be between 0 and 1, where 0 signifies no emotional impact and 1 corresponds to full emotional impact.
The weight for time is calculated in the following way:

$$W_{time,i} = (1 - \frac{1}{1 + \mu \times time} \times max((1 - \lambda + \lambda \times E_{time}), 0)$$

where μ is a variable that is set to a value that fits the time span used in the simulation. time is the emotional effect delay time calculated as:

$$time = L_i + \frac{U_i - L_i}{2} \times ((1 - \sigma \times E_{opt})$$

where E_{opt} is the emotional impact on optimism.
The weight for the planning is calculated as:

$$W_{plan,i} = (1 - \frac{1}{1 + \omega \times plan_i} \times max((1 - \phi + \phi \times E_{plan}), 0)$$

where ω is to fit the planning amount of the simulation.

7. The overall weight of a child node i is calculated as:

$$W_i = \alpha \times W_{Risk,i} + \beta \times W_{Time,i} + \gamma \times W_{Plan,i}$$

The constants α, β and γ give importance to their respective factors.
To select which child node to execute, we list them in ascending order according to the weight value W_i. Hence, the lower value of W_i, the more desirable is the node.

5.1. VRET-Companion Behavior

There are different ways in which the the VRET-character might interact with the user character. Here we would like to show how the emotional selector can be used to let the VRET-Companion choose the behavior which is most suitable under the current emotional circumstances. We design a simple interaction scenario where the character has the following simple interaction choices: it can simply greet the user 'say hi', then it can go away; it can check the weather outside, if there is sun, comments the weather; it can decide to start a conversation, or even play music and start to dance encouraging the user to mimic the dance movements. The character should lay down to rest when its energy is low; and it should maybe go around looking for users. The emotional behavior tree used for the example is depicted in Figure 2. In the tree there is on emotional selector with four child nodes (two sequence selectors s1, s2, and two simple action

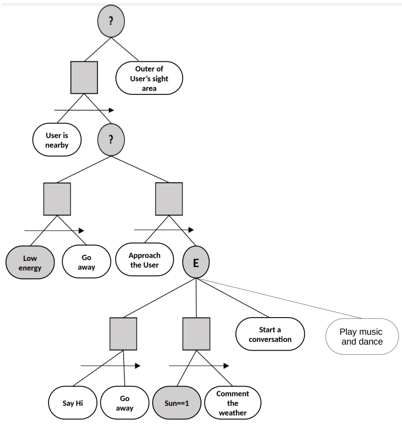

Figure 2: The behavior tree for the VRET Companion

	Say hi	Go away	sun==1	Comment weather	Start a conversation	Play music and dance
Time	[0,0]	[0.3,1.0]	[0,0]	[1.5,2.5]	[7,10]	[2,5]
Risk	0.033	0.0	0.0	0.0	0.7	0.6
Plan	1	1	0	1	1.7	1.9

Figure 3: Time, risk and planning values of leaf nodes

Fear				
	W_{risk}	W_{time}	W_{plan}	W
s1	0.0627	0.034	0.22	0.3167
s2	0	0.0615	0.15	0.2115
n3	1.33	0.087	0.152	1.569
n4	1.14	0.0737	0.213	1.3

Figure 4: Weights For S1, S2, n3 and n4 when the value of fear is 1.0 and The VALUE OF THE remaining emotions is 0.0.

Sadness				
	W_{risk}	W_{time}	W_{plan}	W
s1	0.033	0.34	0.22	0.593
s2	0	0.615	0.15	0.765
n3	0.7	0.87	0.152	1.722
n4	0.6	0.737	0.213	1.55

Figure 5: Weights For S1, S2, n3 and n4 when the value of sadness is 1.0 and The VALUE OF THE remaining emotions is 0.0.

Fatigue				
	W_{risk}	W_{time}	W_{plan}	W
s1	0.033	0.034	0.55	0.617
s2	0	0.0615	0.375	0.437
n3	0.7	0.087	0.38	1.167
n4	0.6	0.0737	0.533	1.2067

Figure 6: Weights For S1, S2, n3 and n4 when the value of fatigue is 1.0 and the value of the remaining emotions is 0.0.

nodes n3, and n4). This emotional selector contains the set of interaction options the character has when it decides to approach the user. The first child node contains a sequence of two actions: to say hi then go away.

Figure 3 shows the amount of risk, planning, and time intervals for every child node of the emotional selector.

For this scenario we consider the emotions: fear, sadness, and fatigue. Constants $\alpha, \beta, and \gamma$ are set to 1.0. And constants $\mu, \lambda, \sigma, \delta, \omega, and \phi$ are set to 0.8, 0.9, 0.5, 0.6, and 0.6 respectively. We use fear as negative emotional impact for risk with a static value of 1.0 as balance since we do not include any positive emotional impact. For planning, we use fatigue as a positive emotional impact. For time, we use sadness as positive emotional impact. For this example we let E_{opt} be zero. The emotions mentioned here are derived from psychological theories presented in Section 5

We let the specified emotions take different values to illustrate the effect this has on the action selection. In Figure 4, 5, 6 we list the overall weights for different factors given different emotional states. It can be easily seen that emotions greatly affect the factor weights in different ways, resulting in different overall weight for the child nodes. The VRET companion example above is simulated under different emotional states. In Figure 7, the weight values for each action are shown under different emotional conditions. It can be seen that the weight values change widely due to emotional impact. For example, when the character is afraid, s2 is the most desirable choice because it is not risky. When the character is sad s1 is the most suitable one because it takes much shorter time to execute. Finally when the character is tired, then s2 is selected since it consists of one action that needs little planning. It is possible to manipulate

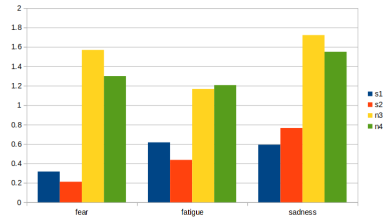

Figure 7: The weight values for the VRET companion example, given different emotional states. When listed, each emotion has the maximum value 1.

the order of the execution of the child nodes to force the character to choose a less desirable node (action) be assigning probabilities to child nodes.

6. Related Work, Conclusions and Future Work

In this work we outlined our line of work on emotional human-automation interaction, with the intention of modeling realistic, believable characters and, more generally, to devise a module for managing emotions in human-AI interaction, to be potentially incorporated in any agent architecture.

A relevant context of empathetic interaction is within synthetic character applications. Several synthetic characters have been developed where empathy and the development of empathic relations played a significant role. These include theatre, storytelling and personal, social and health education (cf., for a survey, to [14]).

Research on computational modeling of empathy has shown that empathic capacity in interactive agents lead to more trust, help coping with stress and frustration and increase engagement [15]. Equipping artificial social agents with empathic capabilities is, therefore, a crucial and yet challenging problem.

Previous existing proposals concerning empathy in agents are discussed in the survey [14] (cf. also the references therein). The approaches discussed in the survey fall within one of the two classes, and, typically, are tailored to a certain application domain. The novelty and relevance of our approach is that: (i) it is fully general (and thus can be exploited in any kind of application); (ii) it encompasses both aspects, as an agent can be an observer that empathize with other agents, particularly with human partners, and at the same time it can lead the user to choose the right course of action.

Currently, we are developing a theoretical framework for modeling emotional empathetic interaction in the context of car interfaces. The research goal is to monitor the emotions of drivers and to enable novel driver-car interactions.

In future perspective, we aim to further generalize our approach and test its applicability in a wider range of contexts, with particular attention to healthcare and teaching, where we intend to deploy solutions and perform practical experiments.

References

[1] R. Calegari, G. Ciatto, V. Mascardi, A. Omicini, Logic-based technologies for multi-agent systems: a systematic literature review, Autonomous Agents and Multi-Agent Systems 35 (2021) 1.

[2] A. Martinich, Philosophy of language: Foundational articles (2009).

[3] S. Costantini, A. Formisano, V. Pitoni, An epistemic logic for modular development of multi-agent systems, in: Engineering Multi-Agent Systems: 9th International Workshop, EMAS 2021, Virtual Event, May 3–4, 2021, Revised Selected Papers, Springer, 2022, pp. 72–91.

[4] A. Dyoub, S. Costantini, I. Letteri, Care robots learning rules of ethical behavior under the supervision of an ethical teacher (short paper), in: Joint Proceedings of HYDRA 2022 workshop and the 29th RCRA 2022 Workshop co-located with the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022), Genova Nervi, Italy, September 5, 2022, volume 3281 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 1–8. URL: https://ceur-ws.org/Vol-3281/paper1.pdf.

[5] P. L. Anderson, M. Price, S. M. Edwards, M. A. Obasaju, S. K. Schmertz, E. Zimand, M. R. Calamaras, Virtual reality exposure therapy for social anxiety disorder: a randomized controlled trial., Journal of consulting and clinical psychology 81 (2013) 751.

[6] M. Iovino, E. Scukins, J. Styrud, P. Ögren, C. Smith, A survey of behavior trees in robotics and ai, Robotics and Autonomous Systems 154 (2022) 104096.

[7] A. J. Champandard, Getting started with decision making and control systems, AI Game Programming Wisdom 4 (2008) 257–264.

[8] B. Knafla, Introduction to behavior trees, Retrieved September 15 (2011) 2011.

[9] A. Johansson, Affective decision making in artificial intelligence: Making virtual characters with high believability, Ph.D. thesis, Linköping University Electronic Press, 2012.

[10] A. Johansson, P. Dell'Acqua, Emotional behavior trees, in: 2012 IEEE Conference on Computational Intelligence and Games (CIG), IEEE, 2012, pp. 355–362.

[11] G. F. Loewenstein, E. U. Weber, C. K. Hsee, N. Welch, Risk as feelings., Psychological bulletin 127 (2001) 267.

[12] J. Lerner, D. Keltner, Fear, anger, and risk, journal of personality & social psychology, 81 (2001).

[13] J. K. Maner, J. A. Richey, K. Cromer, M. Mallott, C. W. Lejuez, T. E. Joiner, N. B. Schmidt, Dispositional anxiety and risk-avoidant decision-making, Personality and Individual Differences 42 (2007) 665–675.

[14] A. Paiva, I. Leite, H. Boukricha, I. Wachsmuth, Empathy in virtual agents and robots: A survey, ACM Transactions on Interactive Intelligent Systems 7 (2017) 1–40. doi:10.1145/2912150.

[15] A. Paiva, F. Correia, R. Oliveira, F. P. Santos, P. Arriaga, Empathy and prosociality in social agents, in: B. Lugrin, C. Pelachaud, D. R. Traum (Eds.), The Handbook on Socially Interactive Agents: 20 Years of Research on Embodied Conversational Agents, Intelligent Virtual Agents, and Social Robotics Volume 1: Methods, Behavior, Cognition, ACM / Morgan & Claypool, 2021, pp. 385–432. doi:10.1145/3477322.3477334.