

FLUMINENSE FEDERAL UNIVERSITY

MARCELO MARQUES DA ROCHA

**Social Robotics Framework and Platform
for Research and Development of
Affective Social Robots**

NITERÓI

2026

MARCELO MARQUES DA ROCHA

**Social Robotics Framework and Platform
for Research and Development of
Affective Social Robots**

Thesis presented to the Computing Graduate Program of Fluminense Federal University as a requirement for the degree of Doctor of Science. Topic Area: Computer Science.

Advisor:

Profa. Débora Christina Muchaluat Saade, D.Sc.

Co-advisor:

Prof. Jesús Favela, D.Sc.

NITERÓI

2026

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

R672s Rocha, Marcelo Marques da
Social Robotics Framework and Platform for Research and
Development of Affective Social Robots / Marcelo Marques da
Rocha. - 2026.
276 f.: il.

Orientador: Débora Christina Muchaluat-Saade.
Coorientador: Jesús Favela.
Tese (doutorado)-Universidade Federal Fluminense, Instituto
de Computação, Niterói, 2026.

1. Robótica. 2. Framework orientado a objeto. 3. Sistemas
multimídia. 4. Produção intelectual. I. Muchaluat-Saade,
Débora Christina, orientador. II. Favela, Jesús,
coorientador. III. Universidade Federal Fluminense. Instituto
de Computação. IV. Título.

CDD - XXX

Marcelo Marques da Rocha

Social Robotics Framework and Platform for Research and Development of Affective
Social Robots

Thesis presented to the Computing Graduate
Program of Fluminense Federal University as
a requirement for the degree of Doctor of Sci-
ence. Topic Area: Computer Science.

Approved in September, 2026.

APPROVED BY

Prof. D.Sc. Débora Christina Muchaluat Saade - Advisor, UFF

Prof. D.Sc. Jesús Favela - Co-advisor, CICESE

Prof. D.Sc. Cláudia Maria Lima Werner, UFRJ

Prof. D.Sc. Alessandra Alaniz Macedo, USP

Prof. D.Sc. Luciana Cardoso de Castro Salgado, UFF

Prof. D.Sc. Igor Monteiro Moraes, UFF

Niterói

2026

Acknowledgments

I would like to thank everyone who stood by my side throughout this journey. Each person played an essential role in helping me reach this point. First and foremost, I thank my God, who motivated me and gave me strength, wisdom, and courage to keep moving forward.

To my wife, Rachel, and my daughter, Luíza, for all their love, companionship, and support throughout this journey. To my parents, for a lifetime of effort and dedication in providing me with the opportunities to become not only a professional, but above all, a good man.

To my advisor, Prof. Débora Muchaluat-Saade, for all her guidance, trust, and the opportunities she provided me throughout this journey. To my co-advisor, Prof. Jesús Favela, for his partnership, collaboration, and friendship. To my friends Marina Ivanov and Rômulo Vieira, for walking alongside me and sharing their experiences, helping me overcome the moments of greatest difficulty inherent to the doctoral journey.

To Universidade Federal Fluminense (UFF), for being my academic home since my undergraduate studies and for playing such an important role in my academic and professional development. To CEDERJ and all the coordinators and professors involved in this project, who, in different ways, contributed to making this journey possible. I would also like to thank CAPES and FAPERJ for their support throughout this journey.

Finally, I would like to express my very special gratitude to Google Research for the remarkable experience and support provided through the Google PhD Fellowship program. To everyone who has been part of this journey, directly or indirectly, my sincere gratitude.

"And God said, Let there be light: and there was light. And God, looking on the light, saw that it was good: and God made a division between the light and the dark"

Gênesis 1:3,4

Resumo

O desenvolvimento de robôs sociais afetivos envolve diferentes desafios relacionados à percepção do estado afetivo do usuário e à expressão dos estados afetivos do robô. Essas capacidades envolvem diferentes mecanismos de *software* que precisam ser integrados às características e capacidades das plataformas robóticas. Esta tese investiga o desenvolvimento de *frameworks* robóticos capazes de incorporar capacidades afetivas. Para isso, é proposto um *framework* de *software* modular e extensível para o desenvolvimento de robôs sociais, organizado em três camadas: *Robot Abstraction Layer*, *Execution Layer* e *Affective Layer*. A *Affective Layer* fornece abstrações para a estimativa do estado afetivo do usuário, a modelagem e adaptação do estado afetivo do robô e seu mapeamento para especificações de comportamento multimodal. Esta tese também apresenta a proposta da plataforma de robótica social afetiva FRED. Além do robô FRED, o *framework* proposto foi concretizado em diferentes plataformas e aplicações, incluindo EVA 2.0, GRACE e FRED MX, incluindo a reutilização do componente da *Execution Layer*, SENSEI nas aplicações *web* e *mobile* EvaPlay e EVA Virtual, respectivamente. A *Affective Layer* foi instanciada e integrada ao robô FRED em dois cenários de interação afetiva. As contribuições desta tese foram avaliadas por meio de diferentes estudos, incluindo a avaliação do FRED com 113 participantes, a avaliação da experiência do usuário em relação à linguagem de *script* EvaML 2.0 e a usabilidade do simulador EvaSIM 2.0 em um *hackathon* educativo, além de uma avaliação qualitativa exploratória da utilização do *framework* por pesquisadores da Suíça e do México, respectivamente, envolvidos nos projetos GRACE e FRED MX. Os resultados fornecem evidências sobre a aplicabilidade do *framework* em diferentes projetos de robótica social, enquanto a instancição da *Affective Layer* demonstrou o funcionamento integrado de seus componentes e a concretização das abstrações propostas. A avaliação do FRED evidenciou diferentes níveis de compreensão dos estados afetivos expressos pelo robô. Em conjunto, o *framework*, a *Affective Layer* e a plataforma de robótica social afetiva FRED, constituem as principais contribuições desta tese para o desenvolvimento e a pesquisa em Interação Humano-Robô Afetiva.

Palavras-chave: Robôs sociais afetivos; Frameworks para robôs; Interação humano-robô; Interação multimodal

Abstract

The development of affective social robots involves various challenges related to perceiving the user's affective state and expressing the robot's own affective states. These capabilities require software mechanisms that must be integrated with the characteristics and capabilities of robotic platforms. This thesis investigates the development of robotic frameworks capable of incorporating affective capabilities. To this end, a modular and extensible software framework for social robot development is proposed, organized into three layers: the Robot Abstraction Layer, the Execution Layer, and the Affective Layer. The Affective Layer provides abstractions for estimating the user's affective state, modeling and adapting the robot's affective state, and mapping these to multimodal behavior specifications. This thesis also presents the proposal for the FRED affective social robotics platform. In addition to the FRED robot, the proposed framework was implemented on different platforms and applications, including EVA 2.0, GRACE, and FRED MX, including the reuse of the SENSEI component from the Execution Layer in the web and mobile applications EvaPlay and EVA Virtual, respectively. The Affective Layer was instantiated and integrated into the FRED robot within two affective interaction scenarios. The contributions of this thesis were evaluated through several studies, including an evaluation of FRED involving 113 participants; an assessment of user experience regarding the EvaML 2.0 scripting language and the usability of the EvaSIM 2.0 simulator during an educational hackathon; and an exploratory qualitative evaluation of the framework's use by researchers from Switzerland and Mexico involved in the GRACE and FRED MX projects, respectively. The results provide evidence of the framework's applicability across different social robotics projects, while the instantiation of the Affective Layer demonstrated the integrated operation of its components and the practical realization of the proposed abstractions. The evaluation of FRED revealed varying levels of understanding regarding the affective states expressed by the robot. Together, the framework, the Affective Layer, and the FRED affective social robotics platform constitute the main contributions of this thesis to development and research in Affective Human-Robot Interaction.

Keywords: Affective Social Robots; Robotics Framework; Human-robot interaction; Multimodal Interaction

List of Figures

1	Overview of the research methodology adopted in this thesis.	19
2	Layered architecture of the Social Robotics Framework. The Affective Layer manages affective perception, emotion modeling, and behavior generation, the Execution Layer manages script execution and orchestration, and the Robot Abstraction Layer provides software abstractions for robotic platforms.	35
3	Main components of the Robot Abstraction Layer.	39
4	Functional organization of the SENSEI execution engine. The figure illustrates the main functional responsibilities involved in EvaML execution, including execution-flow management (Script Player), command execution (Script Node Processor), execution-context management (Robot Memory), external-state synchronization (Shared State Sync), and execution modes. .	41
5	FSM of the SENSEI, illustrating the execution lifecycle from script loading and initialization to script execution and blocking synchronization.	42
6	Functional organization of the SENSEI Script Execution Engine.	46
7	UML sequence diagram of the synchronization of externally generated affective and behavioral states with the SENSEI execution context. PULSE, ROSE, and AEB publish their messages through the MQTT broker, while Shared State Sync receives the subscribed topics and updates the Robot Memory instance maintained by SENSEI.	48
8	Activity diagram of the Script Player execution flow implemented by the <code>play_next()</code> method.	50

- 9 UML sequence diagram of the Script Node Processor functionality. The diagram illustrates how the Script Player retrieves the corresponding *CommandHandler* and delegates node execution. Depending on the node semantics, the handler may access Robot Memory, communicate with simulator components or the physical robot, or evaluate control-flow information before returning the next XML node reference. 52
- 10 An EvaML script being executed by SENSEI in terminal mode (Linux terminal). The terminal displays a large amount of information. Elements of the `<settings>` session can be seen executing at the top of the image, while the outputs of the script's commands appear at the bottom. 53
- 11 Simulation components controlled by SENSEI during EvaML script execution. Items (1) and (2) represent body simulations of two different robot platforms. Items (3) and (4) correspond to reusable execution components that simulate an audio player and a smart bulb, respectively. Items (5), (6), and (7) provide alternative text-to-speech components, including local synthesis (Piper-TTS), a message box displaying the text to be spoken and cloud-based synthesis using the IBM Watson API, respectively. 54
- 12 Architecture and execution pipeline of PULSE. Components with thicker borders indicate framework extension points. 57
- 13 Finite-state machine (FSM) model representing the lifecycle transitions of multimodal input modules. 59
- 14 Multimodal input modules (TER and FER) independently collecting and processing modality-specific data and each producing a single observation for the PULSE at the end of the Affective Perception Window. The probability values shown in the figure's bar charts are for illustration purposes only and do not represent plausible values. 61
- 15 Sequence diagram illustrating the message-based coordination between SENSEI, affective input modules, and PULSE during an Affective Perception Window. 63
- 16 Class diagram of the PULSE extension points for confidence estimation, CAT-to-VAD conversion, and multimodal fusion. 64
- 17 Overview of the ROSE architecture, including the robot's affective profile parameters and its extension points. 66

18	UML state machine diagram representing the dynamics of the robot's affective state in ROSE.	67
19	UML class diagram of ROSE showing the extension points for Empathic Transformation and Emotional Decay.	69
20	Overview of the AEB component architecture. The Empathic Behavior Mapping extension point is highlighted in the figure.	71
21	UML class diagram of the AEB component and its extensible Empathic Behavior Mapping.	72
22	UML class diagram of the communication architecture in SENSEI. The CommunicatorFactory selects the communication strategy defined by the <i>commMode</i> attribute, while all communicators implement the common CommunicatorInterface.	74
23	UML sequence diagram illustrating constructor-based dependency injection in SENSEI. Communication objects are created by the CommunicatorFactory and injected into CommandHandler instances during module loading.	74
24	UML sequence diagram illustrating the dynamic loading of command modules in SENSEI using the Microkernel/Plugin architectural approach.	75
25	Relationship between the extension points proposed in the Affective Layer architecture and the examples of concrete methods presented for each component.	80
26	Mapping of Ekman's six emotions to the VAD space and the coordinates of the anchor points for each. The Neutral emotion corresponds to the coordinate (0, 0, 0) (DE BRUYNE; DE CLERCQ; HOSTE, 2020).	82
27	2D projection of anchor points in the VA plane for the emotions of anger, happiness, sadness, and neutral, and the intensity level points (L1–L4) defined for this instantiation.	85
28	Scenario 1 - Interacting with an "empathetic" affective robot.	89
29	Scenario 2 - Interacting with an "optimistic" affective robot.	92
30	Software architecture of the EVA 2.0 implemented from the proposed framework.	98

31	Controlling the robot's gaze expressions by publishing the message "HAPPY" in the topic <i>evaEmotion</i>	99
32	Instantiation of the Display module.	100
33	Instantiation of the CV (Computer Vision) module.	101
34	Instantiation of the TTS, Audio and Script Player modules.	105
35	Sequence diagram for the execution of the EVA robot speech feature. . . .	107
36	Architecture of the new EVA 2.0 robot software based on the instantiation of the framework proposed in this work.	108
37	Web application to control the modules of the new EVA software.	109
38	Four expressions available in the original version of the EVA robot.	110
39	Eight new expressions proposed with a new design for the EVA robot. . . .	110
40	The three steps of the Computer Vision module execution process.	112
41	Facial recognition process using the Image Embeddings technique.	113
42	GRACE robot. (A) Front-facing image and (B) with a test user (VINAY; UETOVA, et al., 2025).	116
43	FRED MX robot. Left: setup shown in the videos used for both experimental scenarios. Right: the two robot conditions – neutral behavior (top) and affective behavior (bottom) (RANGEL et al., 2025).	118
44	Terminal emulation of the EvaPlay software. A case of reusing the SENSEI component (Execution Layer).	120
45	Image of the Virtual EVA running on a smartphone (MOREIRA et al., 2026). Another case of reusing the SENSEI component (Execution Layer).	121
46	New EvaSIM 2.0 simulator interface and its regions. (1) EvaSIM control bar. (2) Terminal emulator. (3) Tables with the simulator's memory maps. (4) Wizard of OZ interface controls.	128
47	FRED robot with its proper proportions and height in centimeters.	134
48	3D models of FRED and some customized elements for the assembly of his female version, Frida, on the TinkerCad platform.	135
49	Hardware components that are embedded inside FRED.	136

50	Electronic schematic of the assembly of the FRED components.	137
51	FRED control system modules.	141
52	Affective expressions of the FRED robot presented to participants across the six video blocks. Each expression is generated through the combination of nonverbal communication elements, facial display, body pose, and RGB LED patterns, modulated according to the emotion class and its associated probability inferred from the text emotion recognition model. The neutral expression (a) represents the absence of emotion.	148
53	Frequency of positive feedback, negative feedback, and improvement suggestions for each of the categories defined for 74 units of analysis. The percentages of the number of units of analysis for each category are also presented.	166
54	Word cloud based on participants' positive feedback, highlighting the most frequent terms (adjectives) used to describe favorable impressions of the robot. Word size reflects frequency.	166
55	Graphics showing the distributions of answers regarding emotions and blocks in which the Mann-Whitney test returned $p < 0.05$, indicating a statistical difference between the answers of the Tech and Non-Tech groups.	170
56	Percentages for answers from technology and non-technology participants to question Q2 (<i>"Did you perceive when the robot expressed sadness?"</i>) for each of the six blocks.	170
57	Percentages for answers from technology and non-technology participants to question Q3 (<i>"Did you perceive when the robot expressed anger?"</i>) for each of the six blocks.	171
58	Percentages for answers from technology and non-technology participants to question Q4 (<i>"Did you perceive when the robot expressed surprise?"</i>) for each of the six blocks.	172
59	Radar charts showing median values calculated from participant answers for each group, Male and Female (left) and Technological and Non-Technological (right).	176

60	Two views (left and right) of the distributions of the medians of the answers of the participants in the Male and Female groups for the semantic item (Apathetic – Responsive).	179
61	Two views (left and right) of the distributions of the medians of the answers of the participants in the Tech and Non-Tech groups for the semantic item (Ignorant – Knowledgeable).	182
62	Two views (left and right) of the distributions of the medians of the answers of the participants in the Tech and Non-Tech groups for the semantic item (Intelligent – Unintelligent).	183
63	Participants performing hands-on activities during the educational hackathon and testing scripts developed in EvaML 2.0 on the robot using the EvaSIM 2.0 simulator.	189
64	UEQ-S results for EvaML 2.0 after initial exposure (Day 1). Mean scores for Pragmatic Quality, Hedonic Quality, and Overall user experience. . . .	195
65	UEQ-S results for EvaML 2.0 after hands-on interaction (Day 3). Mean scores for Pragmatic Quality, Hedonic Quality, and Overall user experience.	195
66	Standard interpretation of SUS scores based on acceptability ranges and adjective ratings, adapted from (BANGOR; KORTUM; MILLER, 2008). . .	196
67	Distribution of participant responses for each System Usability Scale (SUS) item. Median and interquartile range, with items differentiated by polarity (positive and negative).	197
68	Seal robot, Paro, interacting with lady with dementia (left) and in a pediatric intensive care unit interacting with an 11-year-old boy (right) (SHIBATA; HUNG, et al., 2021).	236
69	Image of the NAO robot (left). Child with ASD interacting with NAO (SHAMSUDDIN et al., 2012).	237
70	Children with hearing disabilities playing the auditory perception game with the robot Pepper (ULUER; KOSE; GUMUSLU, et al., 2023).	238
71	Patricc robotic platform. Mechanical structure (a). Puppet costumes (b-d) (LEVINSON et al., 2021).	238
72	Robot TINY (Teach, Inspire, Nurture, Yield) (JOBALIDA et al., 2024). . .	239

73	EVA Robot Hardware Components.	241
74	Software architecture of the EVA robot.	242
75	A script on VPL.	243
76	Tree structure of an EvaML document.	244
77	Association between EvaML language elements and EVA robot functionalities.	249
78	EvaSIM architecture overview.	249
79	JSON structure of a VPL script (a) and an EvaSIM code representation example (b)	251
80	EvaSIM - Graphical User Interface elements.	252
81	Facial expression capture (userEmotion) and voice capture (listen) simulation windows.	253
82	3D elements that make up the support for attaching a PiCamera to the EVA 2.0 head.	255
83	Mounted support (1) and profile (2) and front (3) views of the support fixed to the EVA head.	255
84	Images of the camera support, its mounting on the robot's head and its connection to the Raspberry PI via the flat cable.	255
85	Main part that connects to the servomotor and the robot body (1). Part for internal fixing of the main part (2). New printed arm capable of holding the SG90 servomotor (3). View of the fittings of the 3 parts (4).	257
86	View of the fittings of the part to the arm (1) and the coupling of these parts to the body (2).	257
87	Real images of the created parts and the SG90 micro servomotor (1, 2, 3, 4). Two printed arms with embedded servomotors and connected parts (5).	258
88	Support piece developed to accommodate the new control board, the Arduino UNO R3 (1). Coupling the support to the EVA base (2).	259

List of Tables

1	Comparison between studies that evaluate the affective expressions of robots. The columns indicate whether the evaluation took into account more than one level of intensity for the emotions, if video-based interaction was used, if the sample was diverse in terms of background, gender and age, and whether the robot expressed emotion using multimodal elements.	27
2	Comparison of robotic software frameworks and development capabilities available in their respective environments. The columns indicate whether the framework is open-source, whether it adopts a distributed architecture, whether a domain-specific language for robot programming and a simulator are available, and whether the framework provides affective support for automatically recognizing, maintaining and generating affective states. . . .	32
3	The main architectural and software design approaches adopted in the proposed framework, their implementation, and their respective objectives. . .	73
4	Topics and messages for the implementation of the EVA 2.0 software. . . .	98
5	Special topics and messages used in the EVA software.	102
6	Comparison between the original EVA platform and EVA 2.0, highlighting the capabilities introduced through the proposed framework.	114
7	Elements and attributes of the EvaML language. Those highlighted in blue result from the extension of the EvaML language, which is one of the contributions of this thesis.	124
8	GQM goals defined for the evaluation of FRED robot.	142
9	Questions for Goal G1.	143
10	Questions for Goal G2.	143
11	Questions for Goal G3.	143
12	Questions for Goal G4.	144

13	Questions for Goal G5.	144
14	Questions for Goal G6.	144
15	Metrics for G1 questions.	144
16	Metrics for G2 questions.	144
17	Metrics for G3 questions.	144
18	Metrics for G4 questions.	145
19	Metrics for G5 questions.	145
20	Metrics for G6 questions	145
21	Mapping of nonverbal communication elements used by FRED for each emotion, based on the probability value of the emotion classified in the Text Emotion Recognition (TER) module in the text.	148
22	Sentences from the complete dialogue between robots FRED and Frida with the indication of the video blocks in which they were spoken and their respective classifications returned by the TER module containing the emotion, with the highest probability and its value.	150
23	Cronbach's α values for each dimension of the GQS and for all items combined.	152
24	Distribution of participants' responses for questions Q1 to Q4 across all video blocks, including expected answers and PEA values, based on all participants ($n = 113$).	153
25	Median values calculated for the answers of all participants, for males and females, for participants in the technological and non-technological areas, as well as the p -values resulting from the non-parametric Mann-Whitney test when comparing the answers of the groups by gender and by professional background.	156
26	Values of the metrics (AT, AN, LK and PI) for all participants and the statistical values resulting from the Mann-Whitney test.	158
27	Values of the metrics (AT, AN, LK and PI) for the Male and Female groups and the statistical values resulting from the Mann-Whitney test.	159

28	Values of the metrics (AT, AN, LK and PI) for the Tech and Non-Tech groups and the statistical values resulting from the Mann-Whitney test. . .	160
29	Value of the PCA metric for all participants, for the groups (Male and Female), for the groups (Tech and Non-Tech) and the statistical values resulting from the Mann-Whitney test.	162
30	Categories and subcategories defined based on participants' open-ended comments regarding FRED's affective expression and its characteristics according to the GQS dimensions.	163
31	Values of the medians of the participants' answers for each semantic item, for each of the groups (Male and Female) and (Tech and Non-Tech, and the statistical values resulting from the Mann-Whitney test.	178
32	GQM goals defined for the evaluation of EvaML 2.0 and EvaSIM 2.0. . .	190
33	Questions for Goal G1.	190
34	Question for Goal G2.	190
35	Metrics for G1 questions.	191
36	Metric for G2 question.	191
37	Descriptive statistics and Wilcoxon signed-rank test results for UEQ-S dimensions (Pragmatic and Hedonic) across Day 1 and Day 3.	194
38	Descriptive statistics and interpretation of System Usability Scale (SUS) scores for EvaSIM 2.0, including acceptability range and adjective rating. .	196
39	Descriptive statistics (median and interquartile range, IQR) of participant responses for each System Usability Scale (SUS) item.	197
40	Summary of responses regarding framework experiences reported by participant P1 (GRACE project) and participants P2 and P3 (FRED MX project).	205
41	Content model of the main sections of an EvaML document.	245
42	Content model of the EvaML elements (commands).	248

Contents

1	Introduction	12
1.1	Motivation	14
1.2	Research Questions	16
1.3	Goals	17
1.4	Research Methodology	17
1.4.1	Ethical Considerations	18
1.4.2	Generative AI Use Statement	19
1.5	Thesis Contributions	20
1.6	Text Structure	21
2	Related Work	23
2.1	Robot Affective Expression Evaluation	23
2.2	Robotic Software Frameworks	26
3	Social Robotics Framework	33
3.1	Architectural Overview	34
3.2	Robot Abstraction Layer	36
3.2.1	Communication Services	36
3.2.2	Synchronization Mechanisms	37
3.2.3	Robot Memory Management	37
3.2.4	Robot Capability Abstractions	38
3.3	Execution Layer	38

3.3.1	SENSEI – Script Execution and Simulation for Empathy Interaction	39
3.3.1.1	Script Execution Lifecycle	42
3.3.1.2	Script Loading and Initialization	43
3.3.1.3	Script Execution Engine	45
3.3.1.4	Execution Modes	51
3.4	Affective Layer	55
3.4.1	PULSE – Processing Unit for Affective Latent State Estimation . .	56
3.4.1.1	Multimodal Input Integration	56
3.4.1.2	Affective Perception Window	58
3.4.1.3	PULSE’s Extensible Affective State Estimation Methods .	62
3.4.2	ROSE – Robot Sentiment Engine	62
3.4.2.1	Dynamic Robot Affective State Model	65
3.4.2.2	ROSE’s Extensible Affective State Transformations	67
3.4.3	Automatic Empathic Behavior – AEB	69
3.4.3.1	AEB’s Extensible Empathic Behavior Mapping	70
3.5	Architectural and Design Patterns	72
3.6	Final Remarks	76
4	Framework Instantiations	78
4.1	Instantiation of the Affective Layer	79
4.1.1	PULSE Instantiation	80
4.1.1.1	Confidence Estimation Instantiation	80
4.1.1.2	CAT-to-VAD Conversion Instantiation	81
4.1.1.3	Multimodal Fusion Instantiation	81
4.1.2	ROSE Instantiation	82
4.1.2.1	Empathy-Driven Transformation Instantiation	83
4.1.2.2	Emotional Decay Instantiation	83

4.1.3	AEB Instantiation	84
4.1.3.1	Empathic Behavior Mapping Instantiation	84
4.1.4	Affective Interaction Scenarios	86
4.1.4.1	Interaction Setup	86
4.1.4.2	Interaction Flow	87
4.2	EVA Robot 2.0	97
4.2.1	Instantiation of EVA Robot 2.0	97
4.2.1.1	Group I Modules	99
4.2.1.2	Group II Modules	101
4.2.1.3	Group III Modules	102
4.2.2	Extending EVA's Nonverbal Communication Capabilities	109
4.2.2.1	New Facial Expressions	110
4.2.2.2	New EVA's Arms	110
4.2.3	Extending EVA's Multimodal Interaction and Affective Capabilities	111
4.2.3.1	Computer Vision Module	111
4.2.3.2	Facial Expression Recognition	111
4.2.3.3	Facial Recognition	112
4.2.3.4	QR Code Reader	112
4.2.3.5	Text Emotion Recognition (TER)	113
4.3	GRACE Robot	114
4.3.1	Research Context	114
4.3.2	Framework Use and Extensions	115
4.4	FRED MX Robot	117
4.4.1	Research Context	117
4.4.2	Framework Use and Extensions	117
4.5	Reuse of Framework Components	119

4.5.1	EvaPlay	119
4.5.2	EVA Virtual	119
4.6	Final Remarks	120
5	Extensions of EvaML and EvaSIM	122
5.1	EvaML 2.0	123
5.2	EvaSIM 2.0	126
5.2.1	EvaSIM - Graphical User Interface	129
5.2.2	EvaSIM - Wizard of Oz	130
5.3	Final Remarks	131
6	FRED Affective Social Robot	132
6.1	Friendly Robot for EDucation and Healthcare: FRED	134
6.1.1	Models for 3D Printing	134
6.1.2	Hardware Components	135
6.1.2.1	Electronic Schematic	136
6.1.3	Robot Capabilities	137
6.1.4	Software Architecture	139
6.2	Instantiation of FRED Control Software	140
6.3	Evaluation of the FRED Robot and its Affective Expressions	141
6.3.1	Methodology	142
6.3.2	Experiment	146
6.3.2.1	Participants and Recruitment	146
6.3.2.2	Instruments and Data Collection	146
6.3.2.3	Experiment Procedure	147
6.3.2.4	Data Analysis	151
6.3.2.5	Reliability Analysis of the Godspeed Questionnaire Series (GQS)	151

6.3.3	Results	152
6.3.3.1	G1 – Affective Expression Capability of FRED	152
6.3.3.2	G2 – Perceived Anthropomorphism	157
6.3.3.3	G3 – Perceived Animacy	157
6.3.3.4	G4 – Perceived Likeability	161
6.3.3.5	G5 – Perceived Intelligence	161
6.3.3.6	G6 – Perceived Child Acceptance	162
6.3.3.7	Qualitative Analysis of Participant Feedback	162
6.3.4	Discussion	165
6.3.4.1	Affective Expression Capability of FRED	165
6.3.4.2	Perceived Anthropomorphism	172
6.3.4.3	Perceived Animacy	173
6.3.4.4	Perceived Likeability	173
6.3.4.5	Perceived Intelligence	174
6.3.4.6	Perceived Child Acceptance	174
6.3.4.7	Semantic Item Exploration	175
6.3.5	Conclusion	182
6.4	Final Remarks	184
7	Evaluation of EvaML 2.0, EvaSIM 2.0 and Framework Applicability	186
7.1	Evaluation of EvaML 2.0 and EvaSIM 2.0	186
7.1.1	Study Design	187
7.1.1.1	First Hackathon Edition (2024)	187
7.1.1.2	Second Hackathon Edition (2025)	188
7.1.2	Methodology	189
7.1.3	Experiment	190
7.1.3.1	Participants and Recruitment	191

7.1.3.2	Instruments and Data Collection	192
7.1.3.3	Experimental Procedure	192
7.1.3.4	Data Analysis	193
7.1.4	Results	193
7.1.4.1	G1 - User Experience with EvaML 2.0	194
7.1.4.2	G2 - Usability of EvaSIM 2.0	195
7.1.5	Discussion	196
7.1.5.1	User Experience with EvaML 2.0	198
7.1.5.2	Usability of EvaSIM 2.0	199
7.1.6	Conclusion	200
7.2	Framework Assessment in HRI Research Contexts	200
7.2.1	Methodology	201
7.2.1.1	Evaluation Design	201
7.2.1.2	Research Questions	201
7.2.1.3	Data Collection	202
7.2.1.4	Data Analysis	203
7.2.2	Participants and Research Project Scenarios	203
7.2.2.1	GRACE Robot	203
7.2.2.2	FRED MX Robot	204
7.2.2.3	Technical Development of FRED MX	204
7.2.3	Results	204
7.2.3.1	Researchers' Experiences with the Framework	205
7.2.3.2	Framework Extensibility	205
7.2.3.3	Framework Strengths and Challenges	206
7.2.4	Discussion	207
7.2.5	Conclusion	208

7.3	Final Remarks	209
8	Conclusions	210
8.1	Answering the Research Questions	210
8.2	Contributions	211
8.2.1	Social Robotics Framework	212
8.2.2	Affective Layer	212
8.2.3	FRED Affective Social Robotics Platform	213
8.2.4	Secondary Contributions	213
8.3	Limitations	214
8.4	Future Work	215
8.5	Publications	216
	REFERENCES	219
	Appendix A - BACKGROUND	235
A.1	Social Robots	235
A.2	Affective Robotics	239
A.3	EVA Robot	241
A.3.1	Robot Architecture	241
A.3.1.1	Hardware Components	241
A.3.1.2	Software Components	242
A.4	EvaML	244
A.4.1	EvaML Document Sections	244
A.4.2	EvaML Language Commands	247
A.5	EvaSIM – EVA Simulator Software	248
A.5.1	The Parser Module	250
A.5.2	The Execution Module	250

A.5.3	Graphical User Interface Module	251
Appendix B - EVA ROBOT 2.0 - NEW 3D PARTS AND MODELS		254
B.1	3D Support for PiCamera	254
B.2	New Arms for EVA 2.0	256
B.3	Support for the Arduino UNO R3 Board	258
Appendix C - RESPONSES FROM THE FRAMEWORK EVALUATION CASE STUD- IES		260

1 Introduction

Social robots have proved effective at improving the physical and cognitive capacities of human beings ([SHIBATA, 2004](#)). Robots have been used in various environments in people's daily lives, becoming assistive devices. In addition to robots that assist humans in some physical activity, such as some type of housework, there are social robots, where the role of the robot is not to provide physical assistance, but rather to provide cognitive stimuli through interactions that resembles human interactions ([SHIBATA, 2012](#)).

The design of social robots is a multidisciplinary field that incorporates the diverse needs of patients and therapists so that these robots can, in fact, be used in healthcare therapies. Designed specifically for therapeutic purposes, the PARO robot (the seal robot) is one of the first social robots to be used in therapeutic applications with the elderly ([BISCHOFF; GRAEFE, 2002](#)). The application of PARO aims to meet the perceived growth in the needs of the elderly, including those requiring long-term care, to receive social and emotional stimulation in a meaningful and positive way. In ([PINO et al., 2020](#)), the social robotics platform NAO was programmed to implement some tasks of the usual protocol of the memory training program, resulting in increased eye contact with patients and reinforcement of therapeutic behavior, reducing, in some cases, depressive symptoms.

Social robots are increasingly being used as a mediator between a therapist and a child in Autism Spectrum Disorder (ASD) therapy studies. Another project with the NAO robot was proposed in ([FUGLERUD; SOLHEIM, 2018](#)). The project was applied with a group of young primary school students with ASD. [Rakhymbayeva, Amirova, and Sandygulova \(2021\)](#) propose a long-term study with 11 children diagnosed with ASD or ASD in co-occurrence with Attention Deficit Hyperactivity Disorder (ADHD). Therapy sessions were customized for each of the children and were based on the children's reactions to the robot's behaviors. Several activities were implemented with the robot and aimed at joint attention, imitation, as well as the emotional well-being of the participants. The results indicated that the proposed therapy was able to sustain the children's engagement.

Demographic and economic factors generate demands for technological support tools. Issues involving schools with limited education budgets, the high number of students per classroom and the need for personalized curriculum for children with diverse needs, stimulate research into technological resources that can assist parents and teachers (BELPAEME et al., 2018). Robots were initially introduced in the educational context as tools to assist in learning programming, facilitating the insertion of students in a technological environment. However, with the advancement of technology, robots have also begun to be used as social agents, being able to play the role of peer, learning companion, tutor or teacher assistant (EKSTRÖM; PARETO, 2022). A study was conducted in (AHMAD; KHORDI-MOODI; LOHAN, 2020) on the effectiveness of using the Cozmo robot in teaching mathematics to a group of students aged 14-17. The participants' subjective assessment of their knowledge in topics such as algebra, geometry and trigonometry improved significantly after the interaction. The results of the use of robots in education suggest an increase in student engagement in the proposed activities as they allow individualized support for each student, also favoring the creation of a more stimulating and challenging learning environment (VITALE; IACONO, 2024).

Affective robotics is a field of study that focuses on how robots can understand and respond to human emotions with the goal of improving the way humans and robots interact (SPITALE; GUNES, 2022). Affective expressions can significantly enhance the ability of robots to interact effectively with humans, making robots appear more socially aware and empathetic. This ability plays an important role in contexts such as healthcare and education, where emotional engagement is an important component of the quality of these interactions (ZHOU; TIAN, 2020).

Robots can be designed to express emotions through multimodal elements. They can express affect using facial expressions, colors (through smart bulbs and RGB LEDs), and static poses (HÄRING; BEE; ANDRÉ, 2011). They can also use movements (EMBLEM et al., 2012), gestures (XU et al., 2013) and sounds (READ; BELPAEME, 2016). These expressions, when designed appropriately, can be adapted to the context and can be adjusted based on the interaction scenario (VASYLKIV et al., 2021). Different communication modalities such as facial expressions and the colors of LEDs, when combined, can convey emotions more effectively. These modalities increase user acceptance of the robot and the effectiveness of communication, especially for complex emotions such as anger, sadness, and surprise (DZHOROEV et al., 2023). In addition, body language can complement other elements of nonverbal communication during the robot's affective expression process, helping the user to identify expressions that may be ambiguous (LI;

ZHAO, 2023).

The ability to express emotions in robots, such as the Pepper robot, has been shown to be effective in helping to rehabilitate children with hearing impairments by accurately conveying emotions such as happiness and sadness, improving the quality of interaction during an audiological rehabilitation therapy session (ULUER; KOSE; OZ, et al., 2020). This same ability to express mood and emotions, as seen in social robot Mini, highlights the importance of affective states in interactions between humans and robots (FERNÁNDEZ-RODICIO et al., 2022).

The adoption of new technologies in education can stimulate the participation of children and young people, while taking care of health can help reduce costs and improve the quality of life of the population. While traditional robotics have long been used in schools and to assist in medical procedures such as surgeries, Social Robots (HENSCHEL; LABAN; CROSS, 2020), which emphasize social interaction with users as their main functionality, have been increasingly developed and adopted. The use of these devices can promote careers in STEAM (Science, Technology, Engineering, Arts, Mathematics), which require hardware and software skills, as well as creativity to program the behavior of the robot.

1.1 Motivation

Robotic technologies are no longer used exclusively on factory floors, but are increasingly present in various environments in people's daily lives. According to Shibata (2012), human-robot interaction has become increasingly common to improve people's quality of life. Robots have begun to act as assistive devices, demonstrating that robotics is a method capable of improving the physical and cognitive abilities of human beings. There are robots, for example, that help with housework, and there are those that assist people with some type of disability in their arms, helping them to feed themselves.

According to Bal et al. (2024), advances in robotics and artificial intelligence have expanded the areas of activity of these robots. Today, they can be found in shopping malls, schools and hospitals. These robots are called Social Robots and are physically embodied and can communicate with humans using social cues by imitating social behavior. These robots are designed to interact with people in appropriate ways in social contexts. Thalmann (2022) states that the emergence of social robots is a phenomenon that dates back to ancient times, reflecting humanity's long fascination with automation

that imitates human life. Since ancient times, the idea of creating mechanical beings that could perform tasks and interact with humans has been a recurring theme in culture and technology. The concept of social robots, as we know it today, began to take shape in the 21st century, with the development of robots that not only perform tasks, but also interact socially with humans. Robots like Nadine ([KANG; BEN MOUSSA; THALMANN, 2024](#)), a human-like robot that can understand and participate in social interactions, exemplify this new era of robotics, representing not only a technological advancement, but also a reflection on the relationship between humans and machines.

The increasing use of social robots in fields such as education and healthcare creates a demand for robots capable of establishing interactions that are more natural and resemble human interactions. In this context, the affective dimension plays a significant role in human-robot interaction ([NAMLISESLI et al., 2024](#)). For a robot to engage in an interaction affectively, two capabilities are particularly relevant: perceiving the user’s affective state and conveying its own affective states in an understandable manner ([SPITALE; GUNES, 2022](#)). These capabilities involve different modalities of perception and expression, as well as their integration into the robot’s behavior during the interaction ([HÄRING; BEE; ANDRÉ, 2011](#)).

Robot software systems need to integrate sensors, actuators, communication mechanisms, and various platform-specific capabilities. In the case of affective social robots, this complexity increases due to the need to integrate mechanisms for perceiving the user’s affect, modeling the robot’s own affective state, and generating behaviors capable of manifesting that state ([SPITALE; AXELSSON, et al., 2025](#)). Adopting suitable software architectures can help manage this complexity by separating responsibilities and fostering modularity, extensibility, and reusability.

Beyond the software challenge, the use of affective social robots in Affective Human-Robot Interaction (AHRI) research also depends on the availability of physical platforms that allow for the implementation and evaluation of these capabilities. Commercial robotic platforms can entail high costs and limitations regarding access or customization ([NAGY; KARL; MOLNÁR, 2024](#)). Open, lower-cost platforms can expand the possibilities for experimentation, replication, and the development of new interaction capabilities, provided they offer mechanisms to adapt both their physical elements and their software systems.

The development of new interaction capabilities also requires suitable mechanisms for specifying and testing robot interactions. Previous work on the EVA robot ([ROCHA; FAVELA; MUCHALUAT-SAADE, 2025](#); [ROCHA; CRUZ-SANDOVAL, et al., 2024](#);

ROCHA, 2022) introduced EvaML, an XML-based language for specifying interactive sessions, and EvaSIM, a simulator for executing and testing EvaML scripts. The new capabilities investigated in this thesis created the need to extend these tools to support the functionalities introduced by the proposed framework.

These challenges motivate this thesis to propose a modular and extensible architecture for affective social robots, incorporating a layer specifically dedicated to affective capabilities, and to extend the supporting tools required for specifying and executing interactions using the proposed functionalities. The proposal aims to connect, in an extensible manner, the perception of the user's affective state, the modeling of the robot's affective state, and the automatic mapping of the robot's affective state to multimodal behavioral specifications, while maintaining a separation between these capabilities and the specific characteristics of different robotic platforms. This thesis also proposes a low-cost affective social robotics platform to serve as a tool assisting researchers in the field of AHRI.

1.2 Research Questions

A social robot's ability to express its affective states constitutes an important aspect of affective human-robot interaction. For these expressions to effectively contribute to the interaction, users need to be able to understand the affective states conveyed by the robot. This is particularly relevant in healthcare and education applications, where affective aspects of the interaction can influence how users perceive and engage with social robots (ZHOU; TIAN, 2020).

During human-robot interaction, an affective social robot should be able not only to convey its own affective state but also to perceive the affective state of the user. Affective information can be obtained from various sources, including facial expressions, speech, text, body language, and physiological signals. These sources can provide complementary information regarding the user's affective state, motivating approaches capable of accommodating and integrating different modalities of affective perception.

Based on these challenges, this thesis addresses the following research questions:

Question 1. *"How can a social robot convey its affective states in a way that is understandable to users?"*

Question 2. *"How can a social robot estimate the user's affective state?"*

1.3 Goals

This thesis aims to propose and investigate a modular and extensible software framework for the development of social robots, with a layer dedicated to the robot's affective capabilities, and also to develop an affective social robotics platform to support research in Human-Robot Interaction.

This thesis has the following specific objectives:

- Design a modular and extensible software architecture for social robots in three layers, separating the control of the specific characteristics of robotic platforms, the execution mechanisms and functionalities related to affective processing.
- Design an extensible affective layer capable of providing abstractions for the estimation of the user's affective state, the modeling and adaptation of the robot's affective state, and the mapping of that state to multimodal behavior specifications.
- Develop an affective, open and low-cost social robotic platform called FRED, capable of expressing affection through different multimodal elements and suitable for research and experimentation in Human-Robot Interaction.
- Extend the EvaML language by adding new XML elements and refactor the EvaSIM simulator to ensure compatibility with the new functionalities introduced by the proposed framework.
- Investigate how the affective states expressed by the FRED robot are understood by users, considering different emotions and intensity levels.
- Investigate an extensible proposal for estimating the user's affective state based on affective information from different multimodal sources.
- Investigate the applicability and use of the developed solutions, through instantiations, functional demonstrations and evaluations on different platforms, applications and research contexts in HRI.

1.4 Research Methodology

The research conducted in this thesis encompasses complementary activities involving literature review, design and development, instantiation and demonstration, and evaluation.

The literature review focused on studies related to the assessment of affective expressions in social robots and on existing robotic frameworks. These activities were guided by the research questions outlined in the previous section and by the requirements identified for the research and development of affective social robots.

The design and development activities resulted in the social robotics framework, the creation of the Affective Layer, and the FRED affective social robotics platform, proposed in this thesis, as well as the software components and tool extensions presented throughout the research. The framework's abstractions and extension points were instantiated into concrete implementations, robotic platforms, and interaction scenarios, allowing for the demonstration of their operation and integration. In particular, the instantiation of the Affective Layer was demonstrated through two affective interaction scenarios executed with the FRED robot.

Different evaluation strategies were adopted based on the characteristics of each contribution and the evidence sought in this thesis. Empirical studies were conducted to evaluate the affective expressions and perceived characteristics of the FRED robot, as well as the user experience with the new EvaML language extension (EvaML 2.0) and the usability of the new refactored version of the EvaSIM simulator (EvaSIM 2.0). Furthermore, the framework's applicability was investigated through an exploratory evaluation based on case studies of independent research conducted by researchers from other institutions. Together with the functional demonstration of the Affective Layer, these evaluations provide evidence to address the research questions and assess the contributions presented in this thesis.

The specific methodological procedures adopted for each study, including evaluation instruments, participants, data collection procedures, metrics, and analysis methods, are described in detail in their respective chapters. Figure 1 presents an overview of the methodology described.

1.4.1 Ethical Considerations

The studies involving human participants presented in this thesis employed consent procedures tailored to the specific characteristics of each evaluation.

For the evaluation of FRED and its affective expressions, conducted with adult participants, individuals provided the Informed Consent Form (ICF) prior to participating in the study. The research protocol was approved by the Research Ethics Committee under

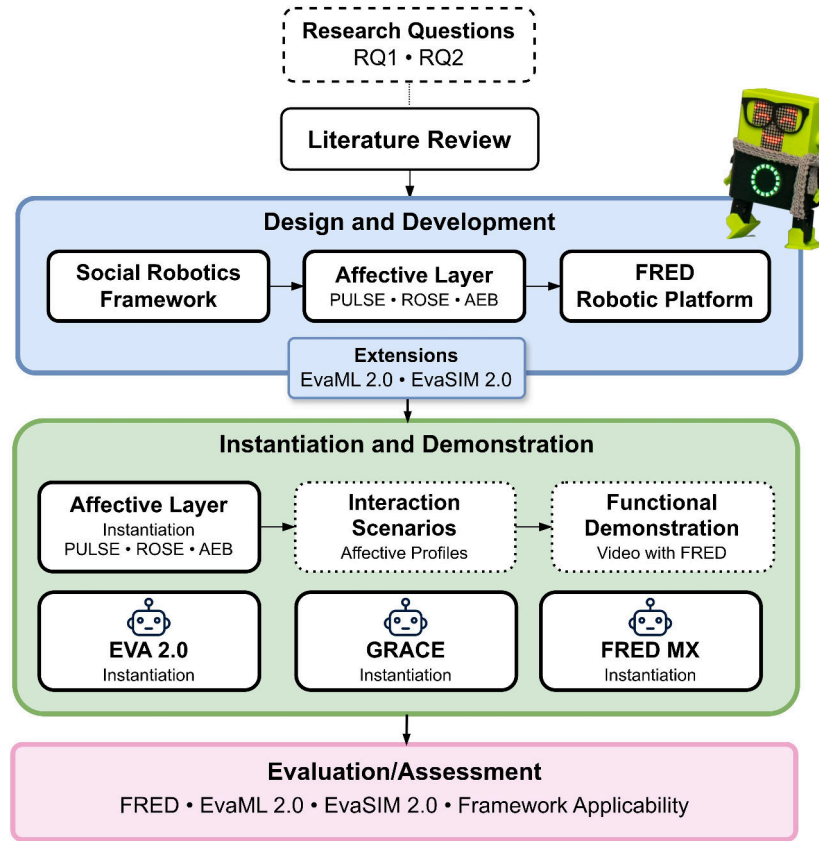


Figure 1: Overview of the research methodology adopted in this thesis.

CAAE number 92434225.1.0000.5243.

For the evaluation of EvaML 2.0 and EvaSIM 2.0, conducted during the hackathon, the study adhered to the guidelines of the Research Ethics Committee of the Fluminense Federal University (UFF). Prior to the start of activities, the ethical documents applicable to research participation were used, including the Informed Consent Form (ICF), intended for participants and/or their legal guardians, where applicable, and the Informed Assent Form, intended for underage participants.

In the exploratory evaluation of the framework involving researchers from the GRACE and FRED MX projects, participants provided the Informed Consent Form (ICF) before completing the questionnaire.

1.4.2 Generative AI Use Statement

During the reading of the articles, the Gemini Notebook was used to assist in the note-taking stage. ChatGPT, utilizing the GPT-5.6 Sol model, supported the translation and revision of the generated text. It was also used to convert Python source code into

the PlantUML¹ graphical language; the resulting figures were reviewed and completely redrawn using the software Inkscape and Draw.io. Any suggestions generated were critically evaluated prior to incorporation, and the author bears sole responsibility for decisions regarding the solution’s development, the conduct of the research, the analysis of results, and the conclusions presented.

1.5 Thesis Contributions

This thesis proposes a modular and extensible Social Robotics Framework for the development of social robot control systems, providing platform-independent abstractions and services for the implementation of independent robotic capability modules. This is the first main contribution of this work.

The second main contribution is proposing an extensible Affective Layer for social robots, integrating mechanisms and abstractions for perceiving the user’s affective state, modeling and adapting the robot’s affective state, and automatically mapping this state into multimodal behavioral specifications through the PULSE, ROSE, and AEB components.

The third main contribution of this thesis is proposing a new affective social robot, called FRED (*Friendly Robot for EDucation and Healthcare*) designed and developed to support research and experimentation in AHRI. This implies designing and building a social robot considering the economic viability and effectiveness of its affective expressions, since it must be able to express itself affectively in an effective way, allowing the user to recognize its affective state.

Three additional secondary contributions emerge from the main contributions presented in this thesis. The first arises from the instantiation of the proposed framework’s Robotic Abstraction Layer, leading to the proposal of version 2.0 of the EVA robot. The second and third secondary contributions stem from new capabilities and features added primarily through the Affective Layer. These contributions enhance the EVA robot’s development tools, specifically the EvaML language and its simulator, EvaSIM, both of which have been extended. The language now includes a broader set of commands supporting new capabilities, such as the integration and use of large language models, the definition of the robot’s affective profile, and more.

¹<https://plantuml.com/>

1.6 Text Structure

This thesis is organized into the following chapters.

To maintain the text concise, the Background Chapter is presented as Appendix A. Appendix A presents the background information necessary to understand this work, as well as an introduction to the EVA robotic platform and the definitions of the EvaML programming language and the EvaSIM simulator, which will be extended throughout the study. The chapter addresses aspects related to social robots and their applications in education and healthcare, as well as affective robotics. Additionally, it briefly describes the functionalities of the upcoming extended versions of the scripting language for the EVA robot, EvaML, and its simulator, EvaSIM.

Chapter 2 presents work related to this thesis. Existing structures for social robots are discussed, highlighting their key characteristics, advantages, and limitations in relation to the structure proposed in this work. The chapter also reviews studies on the evaluation of affective expressions in social robots and compares the evaluation approaches used in them with the one adopted in this thesis.

Chapter 3 presents the proposed framework, with its components divided into the Robotic Abstraction Layer, the Execution Layer, and the Affective Layer.

Chapter 4 presents examples of instantiating the affective component by concretizing extension points abstracted in the definitions of the PULSE, ROSE, and AEB components. A video of an affective interaction between the robot and a user is presented as a functional demonstration of the affective component instantiation proposed in the chapter. Additionally, it introduces three robotic platforms (EVA 2.0, GRACE, and FRED MX) built by instantiating the Robotic Abstraction Layer, as well as two applications (EvaPlay and EVA Virtual) that demonstrate the reuse of the SENSEI component, which belongs to the framework's Execution Layer.

With the addition of the Affective Layer and all the new features and controls it provides, it became necessary to extend the EvaML programming language and the EvaSIM simulator. Chapter 5 presents these improvements.

Chapter 6 presents FRED, the proposed open-source affective social robotics platform. The chapter motivates the robot's construction, a low-cost robot designed for healthcare and education applications in developing countries, as well as its form, movements, and colors. Its hardware characteristics and functionalities are presented. Its software serves

as another example of an instantiation of the framework’s Robotic Abstraction Layer. Furthermore, the chapter evaluates FRED’s affective expression capabilities and robotic characteristics.

Chapter 7 presents the evaluation of the extended versions of EvaML and EvaSIM, as well as an analysis of framework instantiations developed by independent research groups. EvaML 2.0 and EvaSIM 2.0 represent contributions of this thesis and are evaluated in terms of user experience and usability through a hackathon involving novice users. Furthermore, the GRACE and FRED MX instantiations are analyzed as real-world use cases of the proposed framework within different HRI contexts.

Finally, Chapter 8 presents the conclusions, limitations, and directions for future work.

2 Related Work

This chapter presents a literature review focused on two topics related to the scope of this thesis: the evaluation of affective expressions in social robots and software frameworks for robots. Relevant studies were identified primarily through searches in the IEEE Xplore and Scopus databases, using search terms related to these two topics. The selected studies are analyzed and compared based on characteristics relevant to the proposals presented in this thesis. Section 2.1 addresses studies on the evaluation of affective expressions in social robots, considering characteristics such as the emotions evaluated, expression intensity, participant profiles, evaluation procedures, and expressive modalities. Section 2.2 discusses software frameworks for robots, considering their characteristics and development capabilities in relation to the framework proposed in this thesis.

2.1 Robot Affective Expression Evaluation

The studies discussed in this section present empirical evaluations of affective expressions produced by social robots, allowing for a comparison of different evaluation approaches and expressive modalities.

Häring, Bee, and André (2011) presented a proposal that explores how humanoid robots can express emotions through the combination of body movements, sounds and the colors in its eyes (LEDs). The study was conducted with the NAO robot and the emotions evaluated were anger, sadness, fear and surprise. The emotional expressions were divided into separate components for individual evaluation of each modality.

Embgen et al. (2012) investigated how robots could express emotions in an abstract and effective way using body movements that combine human, animal and robot-specific social cues. The study used the Daryl robot, which has movable "ears" and LEDs on its chest in order to express emotions without relying on detailed facial expressions.

With the aim of conducting a study on how robots could express their mood states

through the modulation of behavioral parameters during the execution of functional tasks, [Xu et al. \(2013\)](#) evaluated how simple movements, such as pointing and waving, can be adjusted to reflect different emotional states (positive, neutral and negative). The authors in [\(BECK et al., 2013\)](#) investigated how children interpret the emotions expressed by a humanoid robot through static poses. The study aims to understand whether children are able to correctly identify the emotions conveyed by these poses and whether the position of the robot's head influences emotional perception. The proposal presented in [\(JEON et al., 2015\)](#) conducts a study on the Romo robot, a robot designed to help children with Autism Spectrum Disorder (ASD) recognize emotions. The robot was programmed to express emotions through facial expressions and voice.

Non-Linguistic Utterances (NLUs) are sounds such as beeps and noises used by robots as social signals in human-robot interaction. Unlike spoken languages, NLUs are designed to sound distinctly "robot-like", similar to the sounds produced by the robot R2D2 from the Star Wars movie. [Read and Belpaeme \(2016\)](#) used the NAO robot to emit synthesized NLUs and evaluated how participants interpreted and categorized the emotions associated with these sounds.

[Koyama et al. \(2017\)](#) investigated the influence of different types of facial expressions in robots (emotional and social expressions) on the creation of affective bonds between humans and robots. The work aims to understand how human-robot interaction can be improved in social contexts by evaluating the impact of three types of smiles: a full and genuine smile, involving both mouth and eyes, associated with positive emotions; a social smile, voluntary, involving only the mouth; and a neutral or negative expression.

An automatic reactive response generation system was proposed by [Li, Ishi, et al. \(2017\)](#). The system combines prosodic analysis and sentiment analysis of text to recognize human emotions and generate automatic emotional responses in a natural way. The robot used in the study is ERICA and can generate emotional responses based on the detected emotions, using different levels of intensity. Evaluations were performed to measure the naturalness and appropriateness of the emotional responses generated by the robot.

The research carried out by [Kayukawa et al. \(2017\)](#) aims to investigate how a robot's emotional expressions can influence human in decision-making, mainly in collaborative or competitive scenarios. The study uses a game to evaluate how emotions expressed through body movements, lights (in the eyes) and voice, can affect human choices.

[Rahman \(2024\)](#) analyzed the influence of dynamic emotional expressions of a humanoid robot, called Baxter, in collaborative assembly tasks with humans. The robot's

emotional expressions are used to improve human-robot interaction and performance in industrial assembly tasks.

With the aim of investigating how different forms of emotional expression affect user perception and communication effectiveness, [Cárdenas et al. \(2024\)](#) used the Pepper robot and explored how it can express emotions using body movements and emojis, improving the interaction between the robot and humans. Combinations of body movements and emojis were implemented in order to express basic human emotions.

[Tatsumi and Okubo \(2024\)](#) investigated how changing the color of a communication robot's clothes, in sync with the emotions expressed during a narration, can affect the user's emotional perception and the transmission of information. The Sota robot was used, a tabletop robot with internal lights and translucent clothes that change color in order to represent emotions.

The study conducted by [Raggioli et al. \(2025\)](#) investigated how a robot's body movements (head, arms, speed, and trajectory) contribute to transparent emotional communication in human-robot interaction. The authors conducted a study using the Sanbot Elf robot at a science fair. The majority of participants consisted of elementary and high school students, as well as the general public visiting the fair, ranging in age from 3 to 74 years old. The research evaluated four emotions under three experimental conditions: congruent (face, LED lights, and movements aligned), incongruent (conflicting signals), and neutral (movements only, with a neutral face and LEDs). User evaluations were analyzed using both discrete (emoji-based) and dimensional (valence, arousal, and dominance) models. The robot's physical displays of emotion were based on fixed behavioral profiles for each emotion, without graduated variation in movement intensity.

The proposals presented by [Håring, Bee, and André \(2011\)](#), [Xu et al. \(2013\)](#) and [Li, Ishi, et al. \(2017\)](#) and the evaluation conducted in this thesis, as will be detailed in Chapter 7, unlike the other works presented in this section, evaluate the robot's affective expressions using intensity levels for each of the emotions evaluated. The analysis of these intensities allows a better understanding of the user's emotional perception.

Video-based assessment makes it possible to standardize the presentation of the robot's expressions, avoiding variations that can occur in direct interactions, where the user interacts in real time directly with the physical robot. This type of assessment reduces the influence of external factors and provides equal conditions for all participants. Just like the proposal of [Tatsumi and Okubo \(2024\)](#), this work proposes an assessment that was carried out using recorded videos of the FRED robot's interaction. The remaining

works used direct interaction.

With the exception of the work by [Read and Belpaeme \(2016\)](#) and [Cárdenas et al. \(2024\)](#), which involved a heterogeneous group of participants in their experiments, the other studies relied on participants from more homogeneous groups, specifically, elementary and high school students in the case of ([RAGGIOLI et al., 2025](#)), and university students in the remaining studies. This thesis presents an evaluation with an experiment that has a heterogeneous group of participants composed of people with different professional background, such as technology, education, healthcare and human sciences.

Some of the aforementioned works evaluate robots' emotions in a unimodal way. The proposal in ([XU et al., 2013](#)) evaluates the robot's emotions by simply observing its body movements. In ([BECK et al., 2013](#)) the evaluation is made based on the robot's static poses. An evaluation based on non-linguistic sounds is made in ([READ; BELPAEME, 2016](#)). The works ([KOYAMA et al., 2017](#); [RAHMAN, 2024](#)) use facial expressions. In ([LI; ISHI, et al., 2017](#)) the evaluation is made by analyzing the robot's voice and in ([TATSUMI; OKUBO, 2024](#)) the evaluation of the robot's emotions is made based on the colors of its clothes. While some studies assess affective expressions using a single expressive modality, this work evaluates the affective expressions of the FRED social affective robot, observing multimodal elements, such as facial expression, body poses and the colors of the LEDs on its chest. The FRED robot is also a proposal of this thesis and will be presented in Chapter 6.

As summarized in Table 1, the evaluation conducted with FRED combines characteristics that appear separately in previous studies, including different levels of affective expression intensity, video-based assessment, a heterogeneous sample of participants, and multimodal expressions.

2.2 Robotic Software Frameworks

The literature review focused on robotic software frameworks relevant to the development of social robots. However, the frameworks discussed in this section were not necessarily designed exclusively for social robotics, since some, such as ROS ([QUIGLEY et al., 2009](#)), provide general-purpose architectures and tools applicable to different robotics domains. In contrast, the Social Robotics Framework proposed in this thesis was specifically designed to support the research and development of affective social robots.

[Siciliano and Khatib \(2016\)](#) relates robot architectures to concepts from software en-

Table 1: Comparison between studies that evaluate the affective expressions of robots. The columns indicate whether the evaluation took into account more than one level of intensity for the emotions, if video-based interaction was used, if the sample was diverse in terms of background, gender and age, and whether the robot expressed emotion using multimodal elements.

WORK	INTENSITY LEVELS	VIDEO-BASED INTERACTION	DIVERSE SAMPLE	MULTIMODAL
(HÄRING; BEE; ANDRÉ, 2011)	✓	-	-	✓
(EMGEN et al., 2012)	-	-	-	✓
(XU et al., 2013)	✓	-	-	-
(BECK et al., 2013)	-	-	-	-
(JEON et al., 2015)	-	-	-	✓
(READ; BELPAEME, 2016)	-	-	✓	-
(KOYAMA et al., 2017)	-	-	-	-
(LI; ISHI, et al., 2017)	✓	-	-	-
(KAYUKAWA et al., 2017)	-	-	-	✓
(RAHMAN, 2024)	-	-	-	-
(CÁRDENAS et al., 2024)	-	-	✓	✓
(TATSUMI; OKUBO, 2024)	-	✓	-	-
(RAGGIOLI et al., 2025)	-	-	-	✓
Our Work (2026)	✓	✓	✓	✓

gineering. However, robot architectures differ from conventional software architectures due to the specific requirements of robotic systems. These systems must interact asynchronously and in real time with uncertain and dynamic environments. The choice of an appropriate framework for developing robotic systems depends on the specific requirements of the project, such as the complexity of the desired robotic behaviors and the integration of sensors and actuators.

ROS (Robot Operating System) (QUIGLEY et al., 2009) is not an operating system in the traditional sense of process management and scheduling. According to Martin et al. (2021), ROS has become a standard framework for robotics software development. It is a flexible, open-source framework that provides communication mechanisms, tools, libraries, and rules designed to facilitate software development for different robotic platforms (ALBONICO et al., 2023). ROS can also be integrated with robotics simulators, such as Gazebo, allowing the development and testing of robotic applications in simulated environments (MENGACCI et al., 2021).

Szücs et al. (2018) proposed a software training program based on the NAO robot. The software was built using the NAOqi framework, which addresses programming requirements such as parallelism, synchronization, event management, and resource man-

agement. Furthermore, the Choregraphe Suite ([SUBEDI; PANDEY; MISHRA, 2021](#)), an IDE provided by SoftBank Robotics for programming the NAO and Pepper robots, integrates with the NAOqi framework and provides a visual programming environment for developing robot behaviors. The NAOqi ecosystem also offers simulation capabilities that allow for testing behaviors using a virtual robot.

The YARP (Yet Another Robot Platform) framework ([METTA; FITZPATRICK; NATALE, 2006](#)) was developed by and for humanoid robotics researchers, who face complex hardware and software stacks. It supports multiple communication protocols and provides flexibility for integrating heterogeneous components.

[Ritschel, Kiderle, and André \(2021\)](#) and [Ritschel \(2022\)](#) present an approach for implementing the movements of the Reeti robot independently and in parallel. The approach uses the URBI (Universal Robot Body Interface) framework ([BAILLIE, J.-C., 2004](#)), which is based on a client-server architecture in which the server runs on the robot and is normally accessed via TCP/IP. The framework includes a scripting language that allows the client to control the robot's joints and access its sensors, camera, and speakers. URBI also provides support for the Webots robot simulator.

The RIDE (Robotic Integrated Development Environment) open-source framework aims to manage the complexity and ensure the reliability of mobile robotics applications ([LÓPEZ; PÉREZ; ZALAMA, 2011](#)). It utilizes the RoboGraph tool, which employs Signal-Interpreted Petri Nets (SIPN) to visually edit, monitor, and debug tasks in real time. The framework adopts a layered architecture with hybrid communication: IPC (Inter-Process Communication) and JIPC (Java IPC).

[Yi \(2018\)](#) proposed a modular, general-purpose software framework for intelligent mobile manipulation robots, implemented on the Toyota Human Support Robot (HSR) platform. The framework is composed of five main functional modules that interact with one another: Navigation, Visual Perception, Manipulation, Human-Robot Interaction, and High-Level Autonomy. The framework was initially tested in a simulated environment and subsequently implemented on the physical HSR platform, being successfully used in the RoboCup 2018 international competition.

[Vahrenkamp et al. \(2015\)](#) proposed ArmarX framework, an open-source robotic development environment (RDE) licensed under the GPL. It facilitates and accelerates the creation of high-level motor and cognitive skills for highly complex robotic platforms, such as humanoid robots. Its architecture is organized into three layers: Middleware, Robot Framework, and Application. The middleware layer is built upon Ice (Internet Communi-

cation Engine) technology, which provides a transparent, distributed, cross-platform, and multi-language communication infrastructure, optimized with shared memory channels for large data blocks. The framework continuously exposes the robot's "inner world" and control flow across all levels of abstraction. ArmarX has its own integrated simulator, known as ArmarX Simulator.

The open-source R4A (Robotics4All) framework, proposed by [Panayiotou et al. \(2022\)](#), enables non-expert developers to rapidly create applications for the robot by abstracting away the complexities of hardware, operating systems, and firmware. It employs a Resource-Oriented Architecture (ROA) that abstracts the robot's physical components and algorithms into universal digital resources. This structure is designed as a meta-middleware that operates atop traditional solutions (such as ROS) and is organized into four fundamental layers: the Low-Level Core Agent, which provides encapsulation; the High-Level Core Agent; the Robot Message Broker and Memory Management, which handles messages and memory buffers; and the Robot Resource Transport, which exposes resources via standard web and IoT protocols such as REST, WebSockets, MQTT, and AMQP.

RTSI (RT System Integration), proposed by [Kato and Ohara \(2024\)](#), is an open-source framework featuring distributed execution. It aims to resolve bottlenecks related to integration, data incompatibility, and operational complexity in robotic systems that utilize multiple heterogeneous middlewares, such as OpenRTM-aist and ROS. The framework abstracts integration procedures into standardized packages managed by shell scripts, dividing the operation into three unified stages: module collection, system build, and system run. Through a simplified graphical user interface (GUI), the operator can select the desired robot and execute complex applications with a single click, without requiring specific technical knowledge of the underlying middlewares. It leverages tools from the ROS ecosystem (such as the RViz simulator).

Aiming to democratize the use of social robots in education, therapy, and healthcare, the Sobotify framework ([HÜBERT; YUN, 2024](#)) enables non-technical users, such as teachers, therapists, and educators, to configure robotic behaviors for their specific needs. Its core feature is a robot-agnostic architecture that decouples application code from hardware, offering a unified API to control diverse platforms ranging from humanoid robots (NAO and Pepper) to simple robotic toys (Cozmo and MyKeepon). To eliminate the need for programming, Sobotify allows users to create and edit projects and interactive content (such as quizzes and dialogues) directly within standardized spreadsheets. To overcome

hardware limitations, resource-intensive tasks, including speech processing (Vosk), facial and emotion processing (DeepFace), pose tracking (MediaPipe), are executed on an external computer. Sobotify includes tools for capturing and automatically synchronizing gestures from video and utilizes the MQTT protocol for lightweight, modular communication between its tools and applications. It also features a simplified simulator (a virtual stickman) for testing interactions.

DLS2 (BERTOL *et al.*, 2026) is an open-source framework providing a distributed, modular, and fault-tolerant robotic software architecture designed specifically for complex, dynamic robots such as quadrupeds and humanoids. DLS2 organizes the system into functional layers (such as Control, Estimation, and Service) that exist conceptually across the ecosystem and are dynamically instantiated. The architecture establishes a separation between hard real-time and best-effort tasks, isolating computational resources to ensure that critical control loops do not miss deadlines. It utilizes containerized microservices to isolate dependencies. DLS2 operates over the DDS protocol, optimizing data traffic and preventing network congestion.

As summarized in Table 2, the analyzed frameworks present different combinations of architectural characteristics and resources for the development of robotic systems. All of them adopt distributed architectures, enabling different functionalities to execute in separate modules or processes. Open-source solutions also predominate among the analyzed works, although proprietary environments, such as NAOqi and the framework that runs on Toyota robots, are also used in the development of applications for social robots.

Regarding development and simulation tools, the comparison considers the availability of domain-specific languages and simulation resources in the development environment associated with each framework, not being restricted to tools implemented as internal components of its architecture. Thus, resources made available through tools integrated into the framework are also considered. URBI provides its own scripting language (URBI Script), while RIDE uses RoboGraph and Signal-Interpreted Petri Nets for high-level task programming. Simulation capabilities are available in several of the environments analyzed, including ROS, NAOqi, URBI, ArmarX, the framework proposed by (YI, 2018), Robotics4All, RTSI and Sobotify. These resources are particularly relevant in research contexts, where robotic behaviors need to be programmed, tested, and refined before or during experiments with physical platforms.

Although the analyzed frameworks offer different combinations of architectural features and resources for robotic software development, none of them brings together all

the characteristics analyzed in Table 2. The proposed framework defines an Affective Layer that provides abstractions for estimating the user’s affective state, modeling the robot’s affective state dynamically, and automatically mapping that state to multimodal behavioral specifications, allowing the methods used in these stages to be implemented according to the requirements of each application. Furthermore, its development environment includes a domain-specific language and a simulation environment designed to support the development, testing, and execution of interaction scripts for social robots within HRI research contexts.

This chapter presented studies related to the evaluation of affective expressions in social robots and in robotics software frameworks. The next chapter presents the architecture of the Social Robotics Framework proposed in this thesis.

Table 2: Comparison of robotic software frameworks and development capabilities available in their respective environments. The columns indicate whether the framework is open-source, whether it adopts a distributed architecture, whether a domain-specific language for robot programming and a simulator are available, and whether the framework provides affective support for automatically recognizing, maintaining and generating affective states.

ROBOTIC FRAMEWORK	OPEN SOURCE	DISTRIBUTED ARCHITECTURE	AVAILABLE DSL	AVAILABLE SIMULATOR	AFFECTIVE SUPPORT
Urbi (BAILLIE, J. C., 2005)	✓	✓	✓	✓	-
Yarp (METTA; FITZPATRICK; NATALE, 2006)	✓	✓	-	-	-
ROS (QUIGLEY et al., 2009)	✓	✓	-	✓	-
RIDE (LÓPEZ; PÉREZ; ZALAMA, 2011)	✓	✓	✓	-	-
ArmarX (VAHRENKAMP et al., 2015)	✓	✓	-	✓	-
(YI, 2018)	-	✓	-	✓	-
NAOqi (SZÜCS et al., 2018)	-	✓	-	✓	-
Robotics4All (PANAYIOTOU et al., 2022)	✓	✓	-	✓	-
RTSI (KATO; OHARA, 2024)	✓	✓	-	✓	-
Sobotify (HÜBERT; YUN, 2024)	✓	✓	-	✓	-
DLS2 (BERTOL et al., 2026)	✓	✓	-	-	-
Our work (2026)	✓	✓	✓	✓	✓

3 Social Robotics Framework

Software systems for robots are inherently complex due to the need to coordinate multiple sensors, actuators, communication mechanisms, and decision-making processes while operating in dynamic environments. Robotic systems must continuously monitor their surroundings, respond to unexpected situations, and execute tasks in real time, often through concurrent and asynchronous processes. Such requirements significantly increase software complexity and make the development of maintainable and reusable robotic applications particularly challenging. The adoption of well-designed software architectures and supporting development frameworks has been widely recognized as an effective strategy for managing this complexity (SICILIANO; KHATIB, 2016). However, no single architectural approach is suitable for all robotic applications, and each solution presents its own advantages and trade-offs.

Challenges become even greater in the context of affective social robots. Beyond traditional robotic functionalities, these systems must integrate multiple capabilities, including multimodal perception, affective state modeling, behavior generation, communication, and hardware control (SPITALE; AXELSSON, et al., 2025; ZHAO; GANGARAJU; YUAN, 2025). These capabilities involve distinct responsibilities and may rely on different methods, technologies, and hardware platforms, creating the need for an architecture that enables their integration while preserving modularity and extensibility.

To address these gaps, this thesis proposes the Social Robotics Framework, a modular software architecture designed to support the research and development of affective social robots. The proposed framework adopts an object-oriented design approach and provides extension points that allow for the use of alternative implementations and the addition of new functionalities to its components. Its extensibility is driven by a microkernel/plugin-based architecture, enabling software modules to be dynamically identified and loaded at runtime without requiring modifications to the execution engine. Distributed communication follows the publish-subscribe paradigm, facilitating asynchronous and decoupled interactions among the framework’s components. The only external service required by

the framework is an MQTT message broker (KOZIOLEK; GRÜNER; RÜCKERT, 2020), which provides the communication mechanism for exchanging information between distributed services and robotic modules.

Software architecture naturally involves trade-offs, as design decisions aimed at favoring certain system requirements may affect others (RICHARDS; FORD, 2020; BASS; CLEMENTS; KAZMAN, 2021). This nature of conflicting choices and compromises is extensively grounded in formal architectural analysis methods (KAZMAN et al., 1998). From this perspective, the architectural decisions of the proposed framework are guided by the specific requirements of affective social robots. Rather than aiming to meet the requirements of robotic systems in general, the framework focuses on providing an architecture suitable for the research and development of affective social robots.

The proposed framework follows a layered architecture that separates responsibilities among layers and components, maintaining interoperability through well-defined interfaces and communication mechanisms. The architecture is organized into three layers, described in detail in the following sections. Section 3.1 presents an overview of the architecture and its layers. Subsequently, the layers are presented from bottom to top, with Section 3.2 detailing the Robot Abstraction Layer. Section 3.3 describes the SENSEI execution environment and the Execution Layer. Section 3.4 presents the Affective Layer and its components: PULSE, ROSE, and AEB.

3.1 Architectural Overview

Figure 2 presents an overview of the framework architecture, which is composed of three layers. At the foundation of the framework lies the Robot Abstraction Layer (RAL). This layer provides a common software contract between the framework and platform-specific robotic implementations. RAL encapsulates communication services, synchronization mechanisms, memory management, and reusable abstractions for robot capabilities, allowing higher-level components to operate independently of hardware-specific details. As a result, the framework can be instantiated on different robotic platforms while preserving the same execution mechanisms.

The middle layer, the Execution Layer, is represented by the Script Execution and Simulation for Empathy Interaction (SENSEI) component, which acts as the runtime environment of the framework. SENSEI is responsible for script interpretation, execution management, command orchestration, context handling, and coordination between the

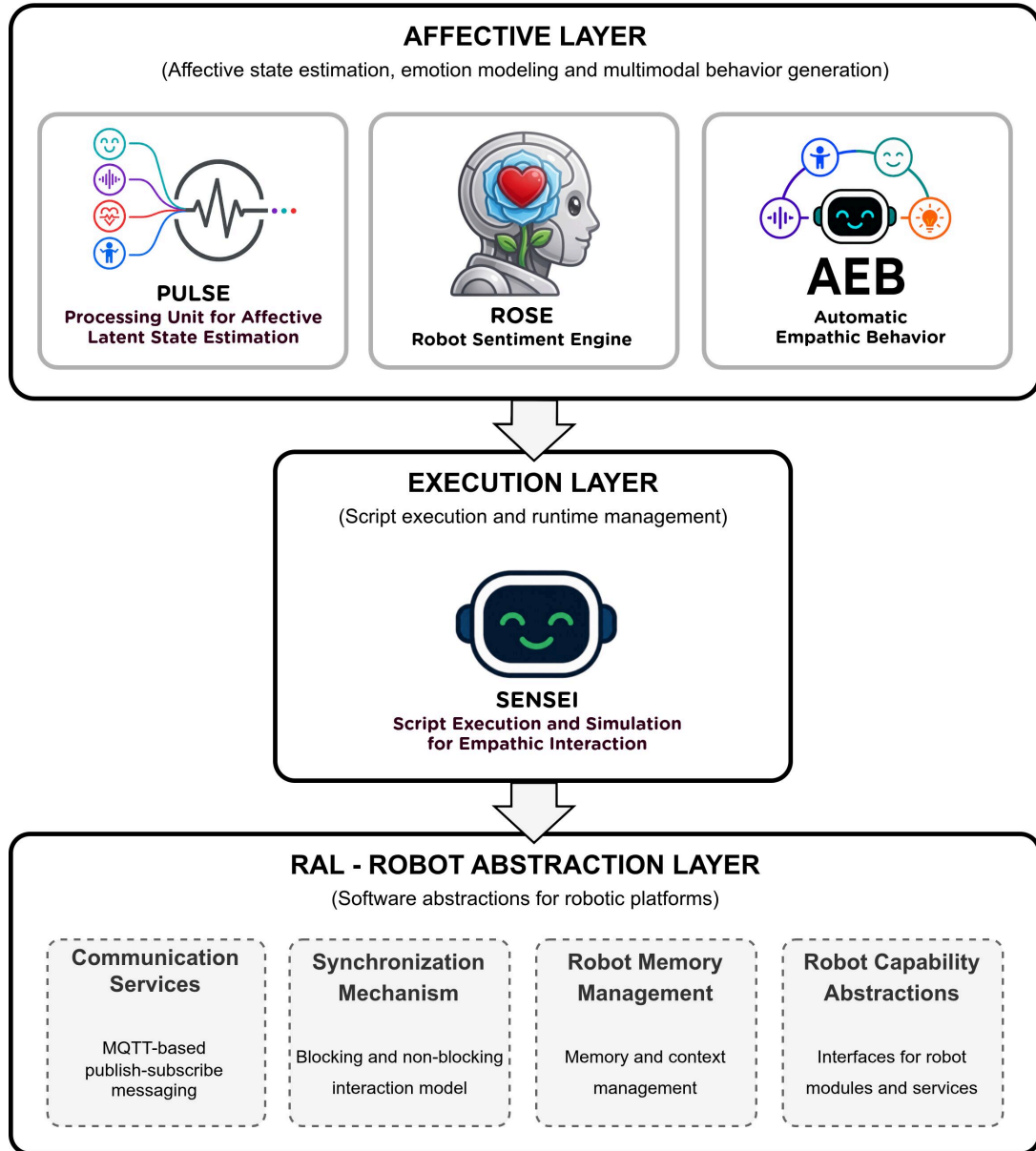


Figure 2: Layered architecture of the Social Robotics Framework. The Affective Layer manages affective perception, emotion modeling, and behavior generation, the Execution Layer manages script execution and orchestration, and the Robot Abstraction Layer provides software abstractions for robotic platforms.

affective components and the underlying robotic services. Through this layer, interaction scripts can be executed in both simulated and physical robotic environments.

At the highest level, the Affective Layer is responsible for coordinating the affect perception, emotional state modeling, and robot behavior generation. This layer is composed of PULSE, ROSE, and AEB components. The Processing Unit for Affective Latent State Estimation (PULSE) performs multimodal affective user’s state fusion and estimation based on information obtained from external perception modules. The Robot Sentiment Engine (ROSE) maintains and updates the robot’s internal affective state, while Automatic Empathic Behavior (AEB) automatically maps the robot’s affective state into multimodal behavioral specifications that can be executed by the robotic platform.

The layered organization adopted in this work provides a clear separation of concerns between affective processing, execution management, and robotic integration. Furthermore, it facilitates the development of new robotic platforms by allowing developers to implement platform-specific functionalities through the abstractions defined by the RAL layer, without requiring modifications to the higher layers of the framework.

3.2 Robot Abstraction Layer

Robotic platforms often differ significantly in terms of hardware, sensors, actuators, communication mechanisms, and software architectures, making it difficult to reuse applications across different robots. To address this challenge, the proposed framework incorporates a Robot Abstraction Layer (RAL) that provides a common set of software abstractions between the framework and the robotic platform. As illustrated in Figure 2, the RAL is organized into four groups of components, Communication Services, Synchronization Mechanisms, Robot Memory Management, and Robot Capability Abstractions, which together provide the software abstractions needed to integrate different robotic platforms into the framework.

3.2.1 Communication Services

The framework adopts a distributed architecture based on the publish–subscribe communication paradigm. This architectural style promotes loose coupling among components and enables modules to execute independently while exchanging information through message passing mechanisms. To support this communication model, RAL provides the `BrokerConnection` class, which encapsulates the configuration and management of con-

nections to the message broker. This component is responsible for storing communication parameters, establishing broker connectivity, and providing access to the messaging client used by robot modules.

By centralizing communication management within a dedicated abstraction, the framework isolates platform-specific networking details from higher-level functionalities, simplifying module development and improving portability across robotic platforms.

3.2.2 Synchronization Mechanisms

Robotic applications frequently involve operations with different execution requirements. While some actions/services can be executed asynchronously, others require synchronization with subsequent tasks before execution can continue. Examples include waiting for the completion of a speech synthesis request, the execution of a movement sequence, or the reception of sensor data. To support these interaction patterns, the RAL defines two synchronization interfaces: `IBlocking` and `IUnBlocking`. These interfaces establish a common contract for modules that require explicit synchronization control during execution.

The `IBlocking` interface defines operations associated with blocking behavior, allowing a module to suspend the progression of an execution flow until a specific task is completed. On the other hand, the `IUnBlocking` interface defines operations that release previously blocked execution flows once the required condition has been satisfied. Together, these interfaces provide a generic synchronization mechanism that allows different robotic modules to implement their own execution logic while simultaneously preserving the cross-platform execution model.

3.2.3 Robot Memory Management

Robotic applications often require the storage of contextual information, execution variables, interaction parameters, and intermediate processing results. To support these requirements, the RAL provides the abstract class `RobotMemory`, which defines a generic memory model for robotic applications. `RobotMemory` includes fundamental memory operations, including initialization, variable creation, value retrieval, and value modification. Abstraction is intentionally independent of any particular implementation strategy, allowing robotic platforms to extend the memory model according to their specific requirements.

3.2.4 Robot Capability Abstractions

The implementation of robotic functionalities within the framework is based on a modular architecture. Rather than defining specific robot capabilities, RAL provides abstractions that establish a common software structure for implementing such capabilities. The `IRobotModule` interface defines the fundamental contract that every robot module must satisfy. This interface specifies the basic lifecycle and communication operations required for interaction within the framework, including module initialization, connection management, message processing, and message transmission.

Complementing this interface, the abstract `RobotModule` class provides reusable infrastructure services that are common to all robot modules. These services include communication support, access to the broker connection, topic management, and event logging. This design allows developers to focus on implementing specific robotic capabilities while reusing the common services provided by the framework.

Figure 3 presents the main components of the Robot Abstraction Layer (RAL). This layer is responsible for the robot's control system as a whole. It does not feature any higher-level abstractions, such as an affective component, but serves as the foundation for implementing such abstractions. Overall, this thesis presents four robotic platforms built through the instantiation and reuse of this layer. Section 4.2, in the next chapter, details its instantiation in the development of the new control software for the EVA robot.

3.3 Execution Layer

The Execution Layer is responsible for coordinating the execution of robot scripts and managing the interaction between the user, the framework components, and the robotic platform. While the Affective Layer estimates and models affective states, the Execution Layer orchestrates the application's behavior by interpreting EvaML scripts. In the proposed framework, this functionality is implemented by the SENSEI component, which acts as the architecture's execution engine. In addition to managing the application's execution context, SENSEI controls affective perception windows, coordinates communication with the PULSE component and multimodal inputs, and sends the multimodal behaviors generated by the AEB for execution on the target robotic platform.

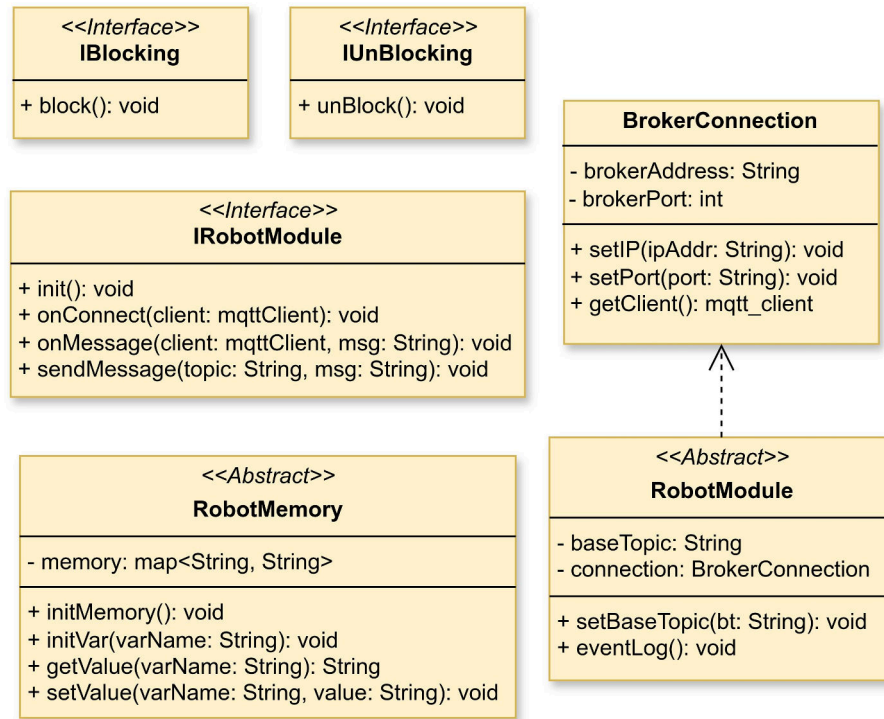


Figure 3: Main components of the Robot Abstraction Layer.

3.3.1 SENSEI – Script Execution and Simulation for Empathy Interaction

SENSEI is the script execution engine of the proposed framework, responsible for orchestrating the complete lifecycle of EvaML applications, from script validation to execution in simulation or on a physical robot. Figure 4 presents its general architecture, organized into four main stages: XML parsing, dynamic module loading, script execution, and execution modes, while the Robot Package provides concrete implementations of EvaML language elements. The script execution process begins with the validation of the EvaML script against its XML schema, followed by the dynamic resolution of EvaML commands into Python modules provided by the Robot Package. The execution engine then controls the application flow, maintains the execution context, and sends commands to the corresponding modules. Finally, the generated actions can be executed either in simulation mode, providing monitoring and debugging information, or on the target robotic platform.

Communication between the framework's components utilizes the MQTT protocol, structured within a topic hierarchy. For standardization purposes, all messages share a base prefix `SIMULATOR_BASE_TOPIC` or `ROBOT_BASE_TOPIC`, accessible via the system variables, the former applying to simulator mode execution and the latter

to robot mode execution. For the sake of notational simplicity, topics are referred to throughout this text in relative terms using only their subtopics (e.g., /PULSE/INPUT), thereby omitting the prefix.

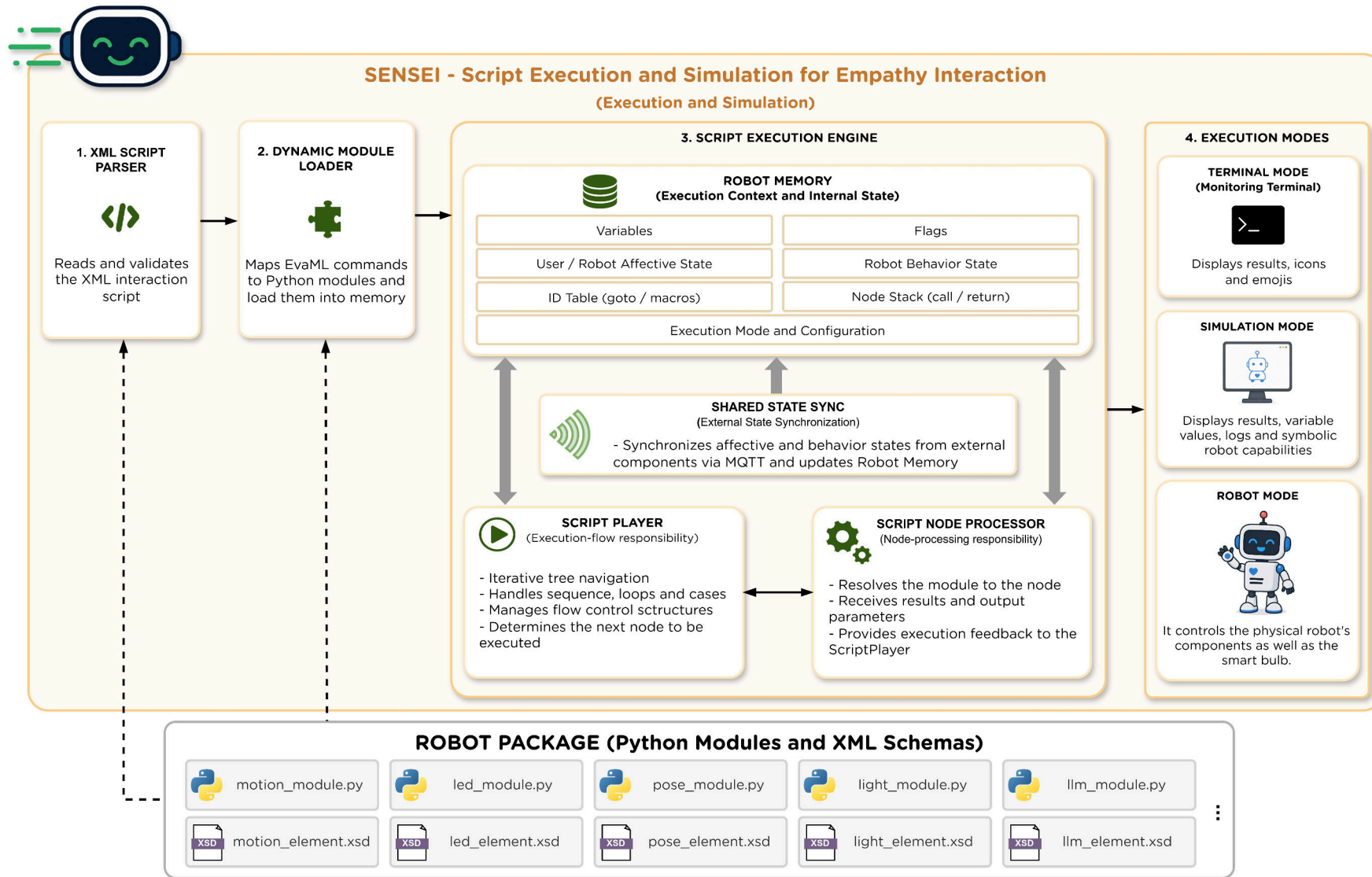


Figure 4: Functional organization of the SENSEI execution engine. The figure illustrates the main functional responsibilities involved in EvaML execution, including execution-flow management (Script Player), command execution (Script Node Processor), execution-context management (Robot Memory), external-state synchronization (Shared State Sync), and execution modes.

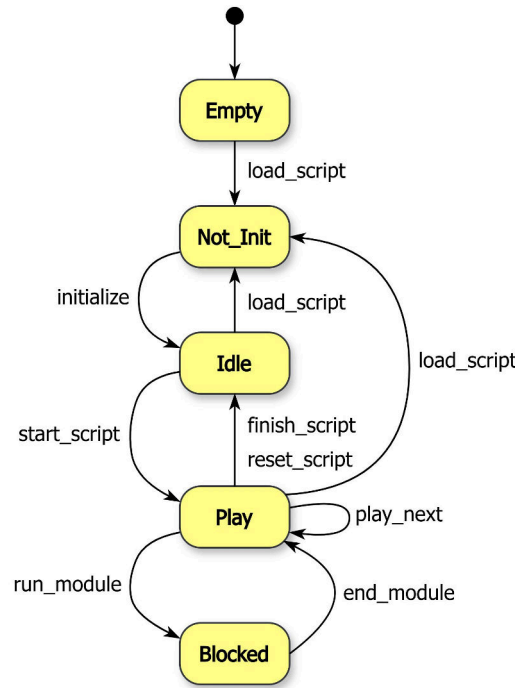


Figure 5: FSM of the SENSEI, illustrating the execution lifecycle from script loading and initialization to script execution and blocking synchronization.

3.3.1.1 Script Execution Lifecycle

The execution of an EvaML application begins with the instantiation of the SENSEI class, which contains the complete execution environment required to process the script. Rather than immediately executing an EvaML application after it is loaded, SENSEI adopts an execution lifecycle controlled by a Finite-state machine (FSM). This design decomposes script processing into well-defined phases, including loading, initialization, execution, and synchronization, ensuring that each stage is completed before the next one begins.

Figure 5 illustrates the Finite-state machine (FSM) that controls the execution lifecycle of a SENSEI instance. The lifecycle begins in the **Empty** state, where no EvaML application has been loaded. Following a successful loading operation, the engine transitions to **Not_Init**, indicating that the EvaML ¹document (XML) has been successfully parsed but the execution environment has not yet been initialized. After the initialization procedures are completed, SENSEI enters the **Idle** state, in which all execution resources have been prepared and the application is ready to start. The sequential interpretation and execution of EvaML nodes take place in the **Play** state. Whenever execution depends on the completion of a blocking operation, such as waiting for a timer, a user response, or the result of an external service, the engine temporarily transitions to the **Blocked**

¹Please refer to Appendix A for more details about EvaML.

state. Once the blocking condition is satisfied, execution automatically resumes in the *Play* state until the script reaches its end, at which point SENSEI returns to the *Idle* state, remaining ready for subsequent executions without requiring the application to be loaded again. The FSM therefore acts as the central coordination mechanism of SENSEI, orchestrating the execution lifecycle independently of the underlying robot platform or execution mode.

This execution model also enables SENSEI to be embedded as a script execution engine within other software applications. Instead of operating only as a standalone application, a SENSEI instance can be instantiated and externally controlled through its public interface, allowing host applications to manage the execution lifecycle programmatically. This capability facilitates the reuse of EvaML execution into different software environments without requiring modifications to the core execution engine. Two examples of this integration through the reuse of the SENSEI component are presented in Section 4.5.

To facilitate understanding of the architecture, the discussion follows the same sequence illustrated in Figure 5, beginning with script loading and initialization, followed by the execution engine and, finally, the available execution modes.

3.3.1.2 Script Loading and Initialization

The first stages of the SENSEI execution lifecycle transform an EvaML application from a static XML document into an executable internal representation by loading the script, dynamically importing the required command modules, constructing the execution metadata, and initializing the execution context. As illustrated by the FSM in Figure 5, this process comprises the transitions from *Empty* to *Not_Init* and subsequently to *Idle*, corresponding to the loading and initialization phases, respectively. During these stages, the XML document is parsed, the execution environment is prepared, the required command modules are dynamically loaded, and the execution metadata is generated. Only after these procedures have been successfully completed does SENSEI become ready to execute the application. The following subsections describe the responsibilities associated with each phase of this process, beginning with the XML Script Parser and then presenting the Dynamic Module Loader and the metadata structures generated during initialization.

XML Script Parser. The process of executing an EvaML script begins with the creation of the SENSEI object. At this point, the object's FSM is in the *Empty* state,

indicating that no XML script has yet been loaded into memory. To proceed, the system must transition to the next state, *Not_Init*. This transition occurs when the SENSEI method `load_script()` is invoked, with the name and path of the script file passed as its parameter.

The `load_script()` method marks the beginning of the script’s execution lifecycle by loading an EvaML interaction script into the runtime environment. During this stage, the XML document is parsed, and its hierarchical structure is transformed into an in-memory tree representation, which serves as the application’s internal model. This process also performs an initial validation of the input, ensuring the script can be successfully read and parsed before any execution-related activities begin. SENSEI transitions from the *Empty* state to *Not_Init*, indicating that the script has been successfully loaded but has not yet gone through the necessary initialization procedures for execution, such as dynamic module loading and preparation of the execution context.

Dynamic Module Loader. After a script has been successfully loaded, the SENSEI remains in the *Not_Init* state until the `initialize()` method is invoked. This stage is responsible for preparing the execution environment by analyzing the script structure and creating all the runtime resources required for its execution. Unlike the `load_script()` method, which is limited to loading the XML document into memory, `initialize()` performs the semantic preparation of the application, ensuring that all execution dependencies are resolved before the first instruction is processed. To ensure that the EvaML language and its SENSEI execution and simulation environment were truly modular and flexible, allowing for the systematic addition of new language elements, it was decided that all elements related to a specific version of the robot would be grouped and stored in a dedicated folder known as `robot_package`. This concept defines a folder containing all files that specify each language element: the XML Schema files defining attributes, data types, and communication protocols for each command, the Python modules implementing the commands’ functionality, simulation elements, and configuration files storing specific settings, such as the base topics used for MQTT message exchange between the physical robot and SENSEI.

The initialization process begins by resetting the execution context stored in the object *RobotMemory*, guaranteeing that variables, execution stacks, and internal control structures from previous executions do not interfere with the new application. Next, the object *ModuleLoader* analyzes the XML tree to identify every EvaML element used in the script. For each distinct element, the corresponding Python module is dynamically

located in the `robot_package` folder, imported at runtime, and instantiated via its *CommandHandler* class. During this process, the object *CommunicatorFactory* creates the appropriate communication object according to the configuration specified by each XML node, allowing the same execution engine to transparently instantiate command handlers for either simulated or physical robot execution.

Once all command modules have been loaded, the object *ScriptMetadata* traverses the XML tree to identify every element containing an `id` attribute. The resulting identifier table establishes direct references between identifiers and their corresponding XML nodes, providing the execution engine with the metadata required to resolve navigation mechanisms, such as `<goto>` instructions and other identifier-based references, during script execution. The initialization phase also stores references to the `<settings>` and `<script>` sections of the EvaML document, enabling the execution engine to separately process the application configuration and its executable instructions.

After all initialization procedures have been successfully completed, the SENSEI transitions from *Not_Init* to *Idle*. This state indicates that the application has been fully prepared for execution: the XML structure has been analyzed, all required command modules have been dynamically loaded and instantiated, the execution metadata has been generated, and the execution context has been initialized. However, no EvaML instruction has yet been executed. Consequently, the *Idle* state represents a stable waiting state in which the execution engine is fully configured and ready to begin script execution when the `start_script()` method is invoked.

3.3.1.3 Script Execution Engine

Once the initialization process has been completed and the execution engine enters the *Idle* state, SENSEI is fully prepared to execute the loaded EvaML application. During execution, the engine repeatedly determines the next XML node to be processed, delegates execution to the corresponding command handler, and updates the execution context according to the outcome of each operation. This functionality is implemented by the Script Execution Engine, whose architecture is illustrated in Figure 6. Its operation is organized around four complementary functional responsibilities: execution-context management (Robot Memory), externally generated information synchronization (Shared State Sync), execution-flow control (Script Player), and EvaML node processing (Script Node Processor).

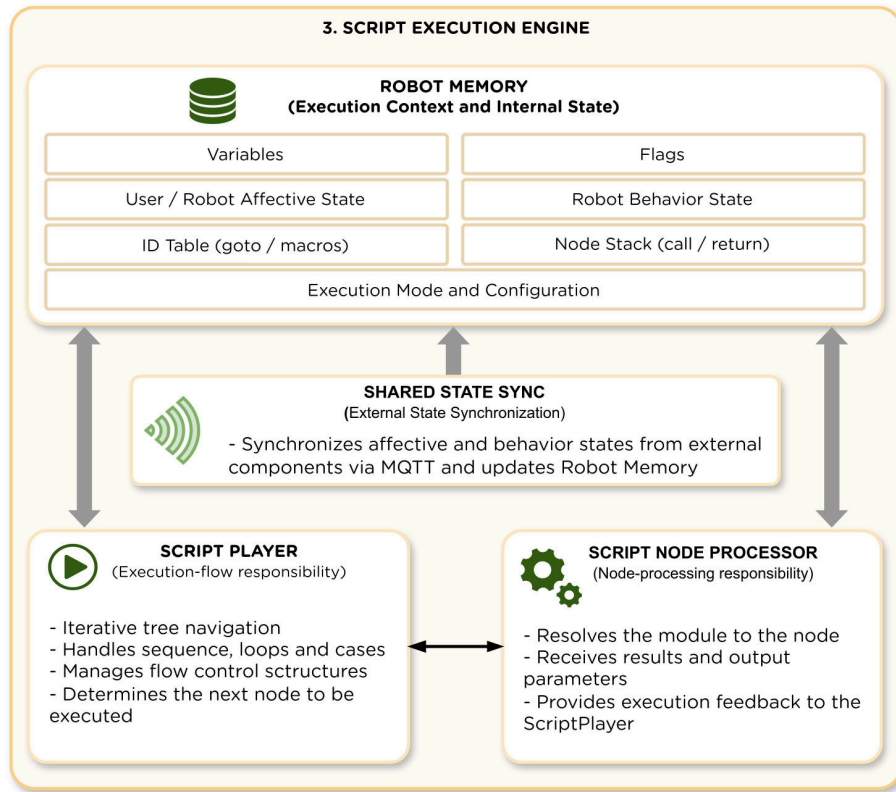


Figure 6: Functional organization of the SENSEI Script Execution Engine.

Robot Memory. The Robot Memory provides the execution context shared by all components of the SENSEI execution engine. An instance of this class is created within SENSEI and passed as a parameter to other objects. As shown in Figure 6, all components can communicate with it, the Script Player and Script Node Processor can execute read and write commands, whereas the Shared State Sync can only write to memory.

Instead of simply acting as a repository for user-defined variables, *RobotMemory* centralizes all information required throughout the execution of an EvaML interaction script, enabling different execution components to exchange data via a common internal state. This shared execution context eliminates direct dependencies between components, allowing each to access or update the information necessary for script execution without maintaining explicit references to the internal states of others.

The execution context maintained by *RobotMemory* comprises various categories of information required during script execution. These include the user-defined variable table, execution flags, the affective states of both the user and the robot, robot behavior state generated by the AEB, the identifier table generated during the initialization phase to resolve `<goto>` instructions and `<macro>` invocations, the node stack used to support control structures, as well as the current execution mode (simulated mode or robot mode)

and configuration parameters loaded from the EvaML application.

By centralizing all runtime information in a single component, the execution engine becomes independent of individual command implementations, while the command handlers remain independent of one another. Consequently, new EvaML commands can be incorporated into the framework without modifying the execution engine or introducing additional communication dependencies between command modules, thereby increasing both modularity and extensibility.

Shared State Sync. Although most of the information required during script execution is generated internally by the execution engine, certain runtime data are continuously produced by components external to it. Examples include the user’s affective state, estimated by PULSE, and the robot’s affective and behavioral states, generated by ROSE and AEB, respectively. Since these components operate independently of the execution engine, their outputs must be synchronized with the execution context to ensure that EvaML commands always operate using the system’s most recent state.

To meet this requirement, SENSEI incorporates the Shared State Sync component, whose primary responsibility is to synchronize externally generated information with the execution context maintained by the Robot Memory. The component continuously receives state updates published by external modules and writes the corresponding values to the Robot Memory, making them immediately available to the Script Player and the Script Node Processor during script execution. This synchronization mechanism is realized through the MQTT protocol’s publish-subscribe communication model. External components publish affective and behavioral state updates to predefined topics, while Shared State Sync subscribes to these topics and updates the corresponding data structures in the Robot Memory. Figure 7 presents a sequence diagram illustrating the role of the Shared State Sync component in an interaction with SENSEI, PULSE, ROSE, and AEB.

This architecture allows asynchronous state changes to be incorporated into the execution process without interrupting the flow of execution, ensuring that EvaML commands always access the most recent affective and behavioral information available within the framework.

Script Player. The Script Player is responsible for controlling the execution flow of an EvaML application. In the current implementation, its responsibilities are concentrated

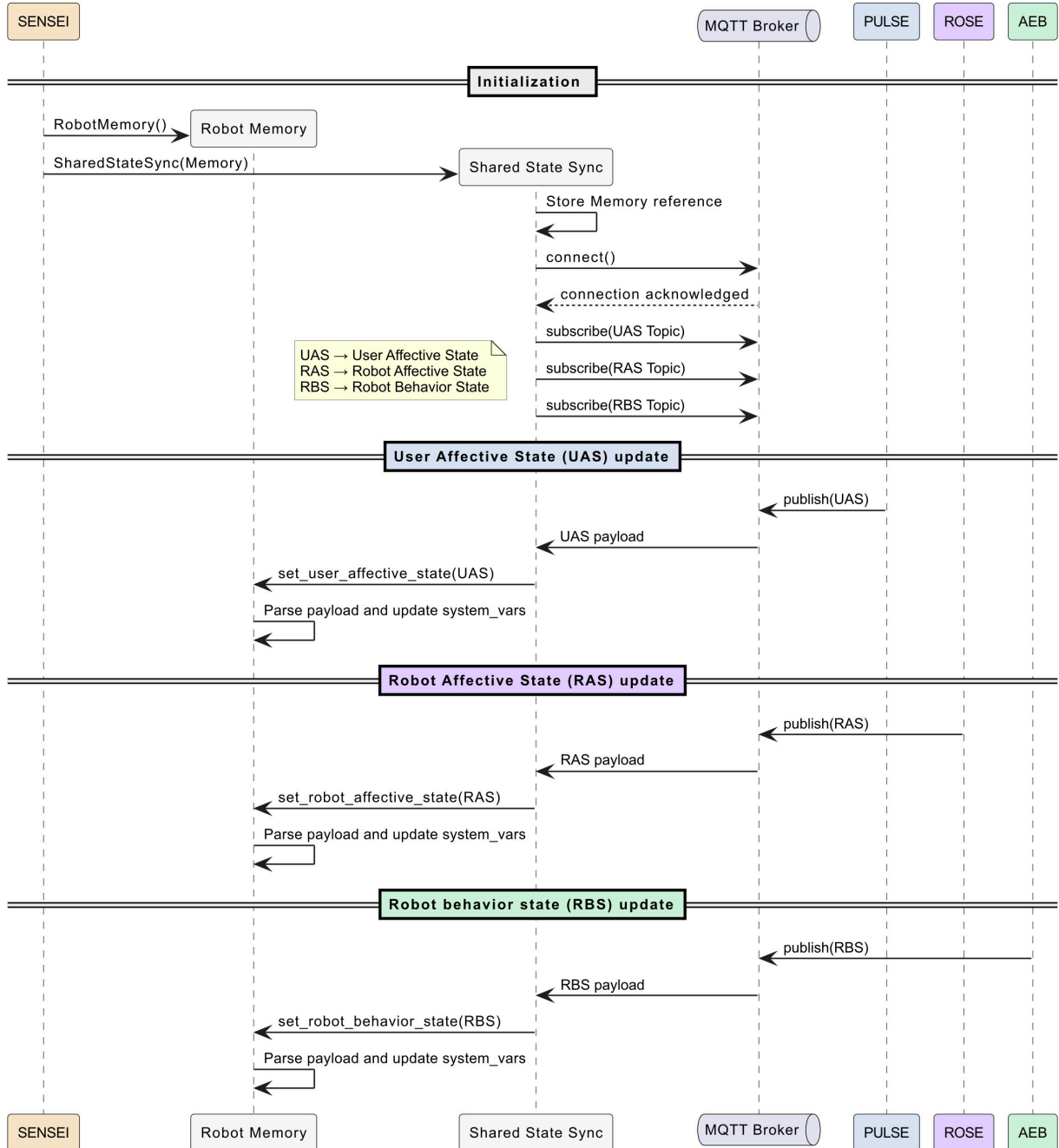


Figure 7: UML sequence diagram of the synchronization of externally generated affective and behavioral states with the SENSEI execution context. PULSE, ROSE, and AEB publish their messages through the MQTT broker, while Shared State Sync receives the subscribed topics and updates the Robot Memory instance maintained by SENSEI.

primarily in the `start_script()` and `play_next()` methods of SENSEI. While the first one prepares the beginning of an execution cycle, the second one determines which XML node must be processed next and updates the current execution position according to the structure and semantics of the script. Therefore, the Script Player acts as the control-flow mechanism of the execution engine, coordinating the execution flow of the EvaML application while delegating the processing of individual commands to the Script Node Processor.

Architecturally, the Script Player separates control-flow management from command semantics. It determines the execution order, manages nested structures and control transfers, and maintains the current position within the XML tree, but it does not implement the behavior associated with elements such as `<talk>`, `<listen>`, or `<light>`. These behaviors are delegated to dynamically loaded command handlers. This separation allows the set of EvaML commands to evolve without requiring corresponding modifications to the main execution algorithm, provided that newly introduced elements conform to the command-handler contract and do not require new control-flow semantics. Figure 8 presents an activity diagram summarizing the algorithm implemented by the Script Player.

Script Node Processor. The Script Node Processor is a functional responsibility of the SENSEI execution engine for executing the semantics associated with each EvaML language element. While the Script Player controls the execution flow by determining the next XML node to be processed, the Script Node Processor interprets that node and performs the corresponding action according to its implementation.

Each EvaML element is implemented as an independent Python module located in the `robot_package` directory. These modules implement a common execution interface through subclasses of `BaseCommandHandler`. Each module defines a `CommandHandler` class that implements the `node_process()` method. This method receives the current XML node and the Robot Memory instance, executes the behavior defined for the element, and in most cases, the current node itself is returned, allowing the Script Player to continue the execution sequence normally. However, control-flow elements, such as `<goto>` and `<useMacro>`, may return a different node, thereby modifying the execution path.

Depending on the element's semantics, processing modules can interact with the Robot Memory to read or update variables, access system configuration parameters, send/receive messages to external components over the network using protocols such as

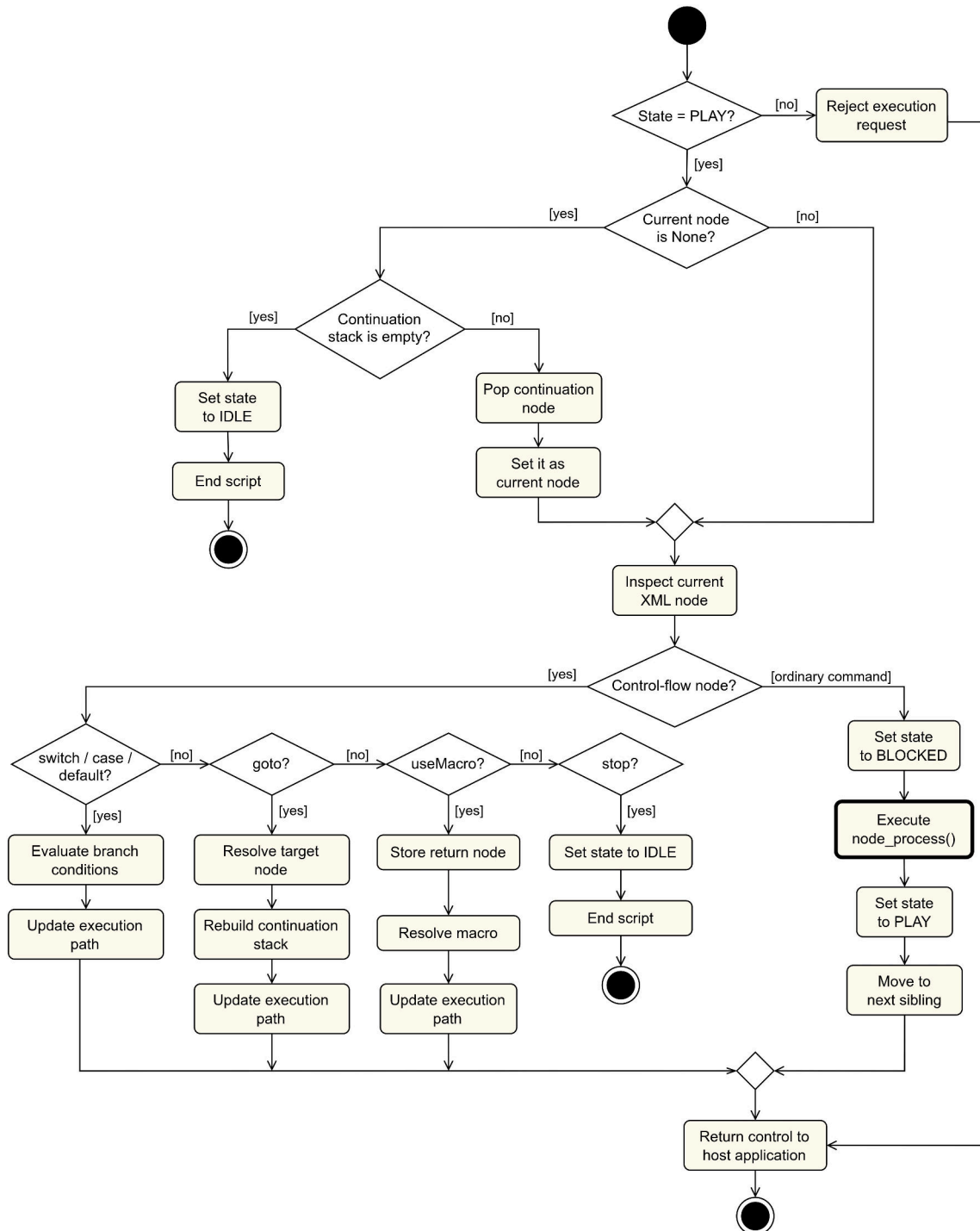


Figure 8: Activity diagram of the Script Player execution flow implemented by the `play_next()` method.

HTTP, MQTT, etc., control simulator or physical robot elements, or evaluate control flow conditions.

This design provides a uniform execution mechanism for all EvaML elements while isolating their implementations into independent processing modules. Figure 9 details the interactions that implement the Script Node Processor functionality. Although not represented as an independent software component, this functional responsibility comprises the retrieval of the appropriate *CommandHandler*, delegation of node execution, interaction with the execution context and communication infrastructure when required, and the return of the node reference used by the Script Player to determine the subsequent execution step.

3.3.1.4 Execution Modes

The SENSEI execution engine supports multiple execution modes that allow the same EvaML application to be executed in different environments without modifying the script itself. The execution mode is selected when the application starts and stored in Robot Memory, making it available throughout the execution. Command handlers use this information whenever their behavior depends on the execution environment. This design separates the application logic from platform-specific details, allowing a single EvaML script to operate in terminal, simulation, or physical robot environments.

Terminal Mode. In Terminal mode, the application runs locally without communicating with simulator components or the physical robot. Commands that typically interact with simulation elements or actuators are executed directly on the host computer. For instance, audio playback occurs locally rather than involving the publication of MQTT messages. Furthermore, user voice input is handled via the keyboard (within the terminal). The robot’s speech is printed on the screen, and various elements, such as the smart bulb and facial expressions, are represented by colorful icons and emojis. Although the terminal interface is somewhat rudimentary, the use of detailed colors and icons makes the visual output quite playful and graphically appealing. This mode is primarily intended for debugging, language development, and rapid testing of script logic, eliminating the need for a simulator or robot. Figure 10 shows a screenshot of a small script running in terminal mode on SENSEI.

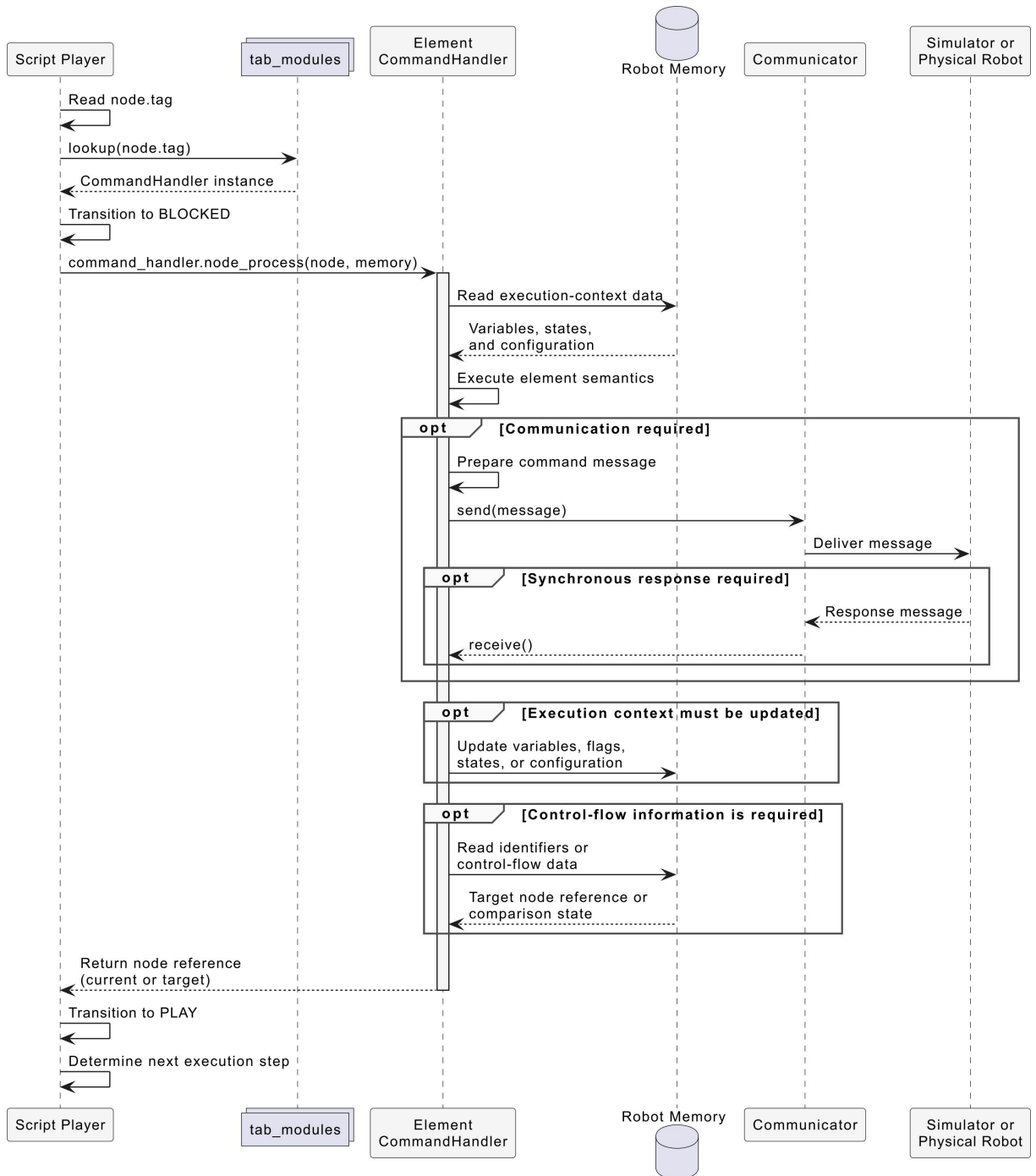


Figure 9: UML sequence diagram of the Script Node Processor functionality. The diagram illustrates how the Script Player retrieves the corresponding *CommandHandler* and delegates node execution. Depending on the node semantics, the handler may access Robot Memory, communicate with simulator components or the physical robot, or evaluate control-flow information before returning the next XML node reference.

```

Starting the script in TERMINAL MODE - Reading the global <settings> from file: fred scripts/framework2_evaml.xml
STATE: Setting the robot default voice=en_US-bryce-medium.
STATE: Setting the light effects=ON.
STATE: Setting the audio effects=ON.
STATE: Setting the robot affective profile=OPTIMISTIC.
STATE: Setting the robot empathy=0.40.
STATE: Setting the robot emotional decay=1.0.
STATE: Setting the robot emotional decay delay=3.0.
STATE: Setting the robot base mood (VAD vector)={"valence": "0.40", "arousal": "0.30", "dominance": "0.20"}.
Executing the script: fred scripts/framework2_evaml.xml
STATE: Setting the Smart Bulb. (IP:192.168.99.11, PORT:55443). ON
STATE: Waiting for 1 seconds. 00:00
STATE: The Robot is speaking the sentence: "Hello, how are you?"
STATE: Using macro START.
STATE: Setting the robot LEDs to the animation/color GREEN!.
STATE: Setting the Smart Bulb. (IP:192.168.99.11, PORT:55443). ON
STATE: Jumping 4 to the element Light with id=POINT1.
STATE: Setting the Smart Bulb. (IP:192.168.99.11, PORT:55443). ON
State: End of script!
Script finished: fred scripts/framework2_evaml.xml

```

Figure 10: An EvaML script being executed by SENSEI in terminal mode (Linux terminal). The terminal displays a large amount of information. Elements of the `<settings>` session can be seen executing at the top of the image, while the outputs of the script’s commands appear at the bottom.

Simulation Mode. In simulation mode, SENSEI communicates with a set of independent simulation components using the same MQTT-based communication interface adopted for the physical robot. Executions in the simulator and on the robot preserve the same message structures and capability-specific subtopic suffixes, while distinct base topics identify the target environment. Command handlers publish messages to the topic specified by the `SIMULATOR_BASE_TOPIC` system variable when operating in simulation mode, and to the topic specified by the `ROBOT_BASE_TOPIC` variable when operating in robot mode. Consequently, the communication logic remains identical across both environments, while the topic prefix routes each command to the appropriate implementation, whether simulated or physical.

Unlike previous simulator versions, named EvaSIM (ROCHA; CRUZ-SANDOVAL, et al., 2024), in which the simulated robot capabilities were integrated into a single graphical application, the current architecture separates them into independent components. This organization allows a component to be added, replaced, or omitted without modifying SENSEI or the EvaML application. It also enables the simulation of a different robot through a new set of components that implement the expected MQTT interface, message structures, and subtopic suffixes under the simulator base topic, rather than requiring the development of another monolithic graphical simulator. The framework specifies that a robot’s simulation components must be implemented and stored in a folder named `sim_components`, located at the root of the `robot_package` folder.

Figure 11 shows simulation elements for two robotic platforms, FRED (a 3D version created in Unity (1)) and EVA (the original 2D version(2)), respectively, demonstrating that changing the simulated robots is very simple in this new simulation approach. Items

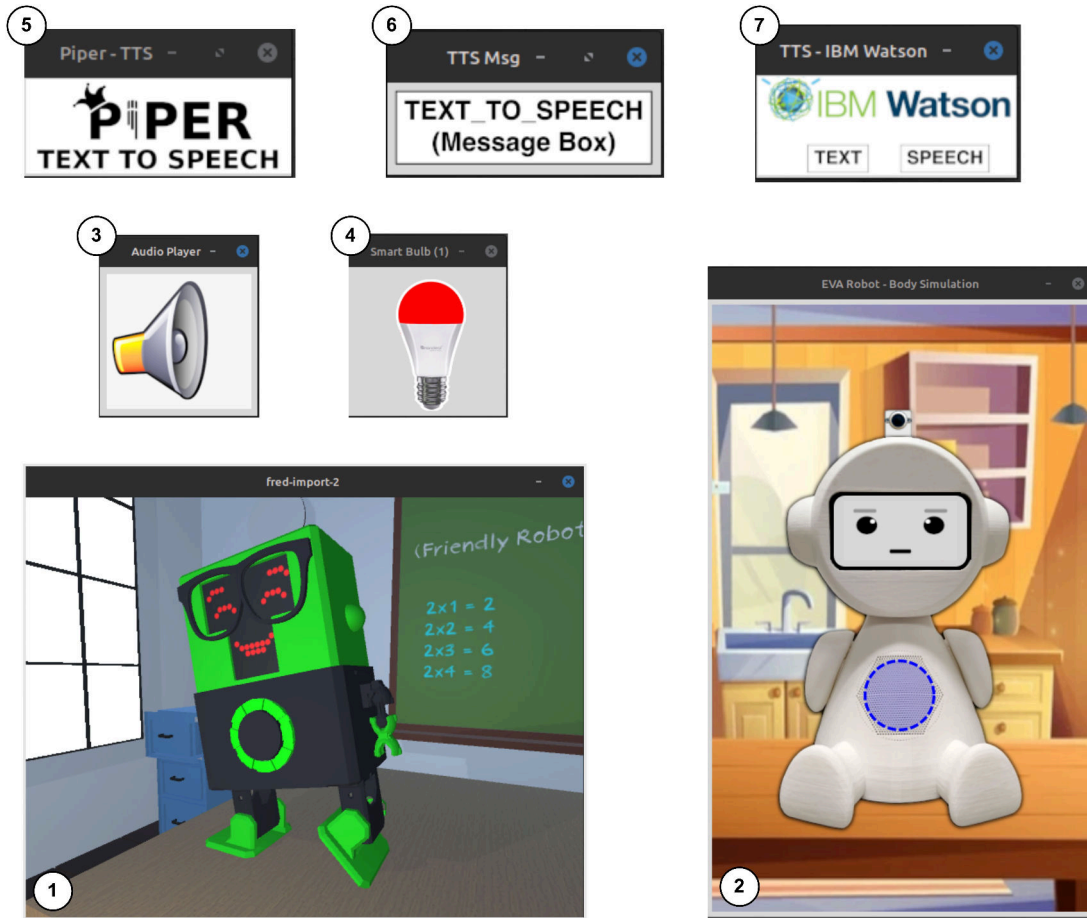


Figure 11: Simulation components controlled by SENSEI during EvaML script execution. Items (1) and (2) represent body simulations of two different robot platforms. Items (3) and (4) correspond to reusable execution components that simulate an audio player and a smart bulb, respectively. Items (5), (6), and (7) provide alternative text-to-speech components, including local synthesis (Piper-TTS), a message box displaying the text to be spoken and cloud-based synthesis using the IBM Watson API, respectively.

3 and 4 represent the audio device and the smart bulb. Items 5, 6, and 7 are elements for text-to-speech (TTS) generation. Element 5 is a local TTS, while element 7 is a cloud-based TTS. Item 6 is a TTS component that converts spoken text, contained in a script, into a message box.

Robot Mode. In Robot mode, the command handlers publish MQTT messages under the robot base topic, allowing the generated commands to be delivered to the physical robotic platform. Robot-side software components subscribe to these topics and execute the corresponding actions, such as speech synthesis, audio playback, lighting control, robot motion, or affective behaviors. Since the communication protocol, message structures, and capability-specific subtopic suffixes are identical to those used in Simulation mode, the same EvaML application can be executed on the physical robot without modification.

This organization preserves a clear separation between the execution engine and the robotic platform. The Script Player, Script Node Processor, Shared State Sync and Robot Memory operate independently of the target hardware, while platform-specific implementations are responsible for interpreting and executing the published commands. Consequently, as it was presented in Section 3.2, supporting a new robot primarily requires implementing the corresponding robot-side components that subscribe to the expected MQTT topics and provide the required capabilities, without requiring modifications to SENSEI or to existing EvaML applications.

This section detailed the framework’s Execution Layer. It presented SENSEI’s components and functionalities, as well as the complete lifecycle of an EvaML application, from the moment the XML file is loaded to its execution in one of the three available modes. Furthermore, it described the functionalities of the Script Player, Script Node Processor, and Robot Memory, along with the process for synchronizing state information and affective behaviors generated by the external components PULSE, ROSE, and AEB, using the Shared State Sync component. The following section addresses the Affective Layer, the highest layer of the framework.

3.4 Affective Layer

According to [Namlisesli et al. \(2024\)](#) and [Williams et al. \(2023\)](#), affective robots can provide emotional support, reduce anxiety and stress, and promote positive emotions, contributing to improved emotional well-being in both educational and healthcare environments. According to [Vashishth et al. \(2025\)](#) and [Mishra, Tripathi, and Kumar \(2024\)](#), they are capable of providing personalized interactions by recognizing and responding to human emotions. This section presents the Affective Layer proposed in this thesis. Its objective is to provide a set of components that can be extended to endow the robot with affective behavior.

The Affective Layer is responsible for providing abstractions for the perception, modeling, and expression of affective information within the proposed framework. While the Execution Layer coordinates the execution of interaction scripts, the Affective Layer distributes the affective interaction process among three complementary components. PULSE provides abstractions for estimating the user’s affective state based on perceptual information, ROSE provides abstractions for dynamically maintaining and transforming the robot’s affective state according to the interaction, and AEB provides the abstraction

for automatically mapping this state to multimodal behavioral specifications.

3.4.1 PULSE – Processing Unit for Affective Latent State Estimation

PULSE is the component responsible for coordinating and processing multimodal information to estimate the user’s affective state. Its design decouples multimodal perception mechanisms from the framework, allowing external input modules to employ different sensing technologies, models, and processing approaches without requiring their implementation to be incorporated into the framework.

Multimodal input modules can work as independent distributed components and communicate with the framework through the message-based communication mechanisms adopted by the architecture. PULSE receives data (observations) from the sources selected for a given affective perception window that processes them. Each source is identified by a unique identifier and provides its observation either as a categorical emotion representation (CAT) or as a Valence-Arousal-Dominance (VAD) representation.

As soon as the data from all sources expected for the current affective perception window have been received, PULSE processes them through three stages: confidence estimation; category to valence, arousal and dominance (CAT-to-VAD) conversion, and multimodal fusion. It is not the framework’s purpose to define implementations for the algorithms used in each stage, instead, these methods are defined abstractly, allowing alternative confidence and CAT-to-VAD conversion methods, and multimodal fusion strategies to be incorporated. The resulting fused VAD vector represents the estimated affective state of the user and is published through the `/USER_AFFECTIVE_STATE` subtopic, making it available to other framework components. Figure 12 presents an overview of the PULSE architecture and execution pipeline, highlighting steps 3, 4, and 5. These serve as extension points within the framework, allowing the confidence estimation, CAT-to-VAD conversion, and multimodal fusion methods to be defined (implemented) by the user.

3.4.1.1 Multimodal Input Integration

Multimodal affective perception can rely on heterogeneous sources, such as facial expressions, speech, text, gestures, or physiological signals. Furthermore, different applications may employ distinct sensing devices, recognition models or processing techniques for the same modality.

The proposed framework does not define how multimodal input modules must ac-

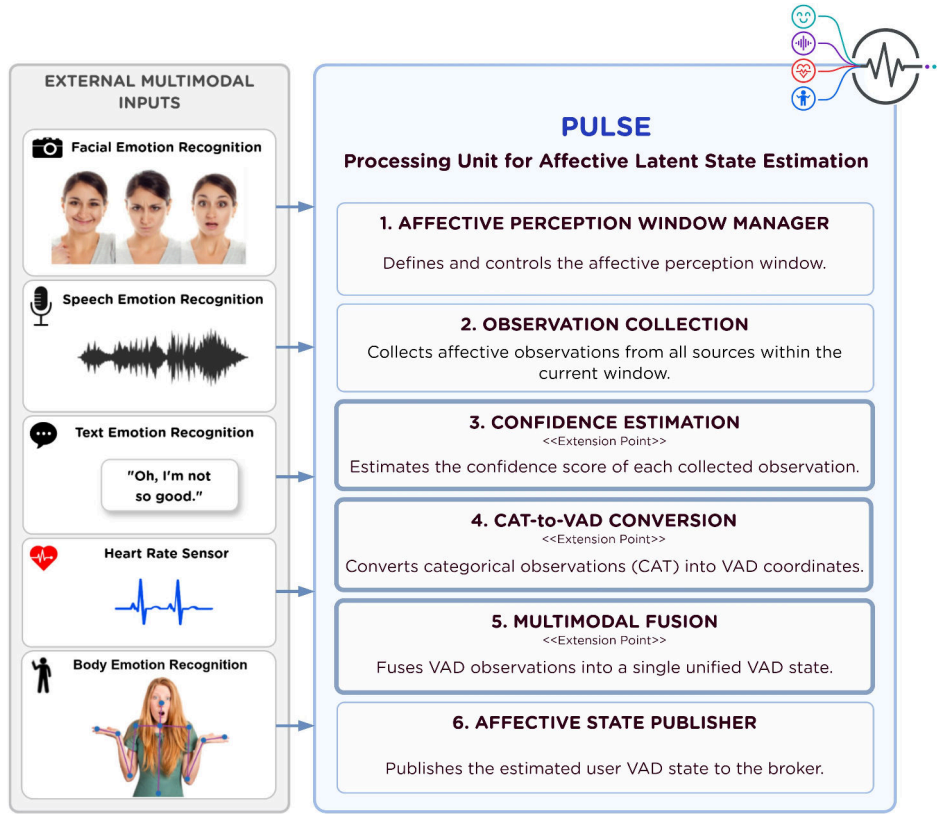


Figure 12: Architecture and execution pipeline of PULSE. Components with thicker borders indicate framework extension points.

quire or process their data. Instead, it introduces PULSE, which proposes a message-based integration mechanism, allowing external modules to participate in estimating the user's affective state without requiring their implementations to be embedded within the framework itself. In this proposal, each multimodal input module operates as an independent component responsible for acquiring and processing its specific data. Furthermore, PULSE defines three abstract classes to be used for implementing three functionalities involved in the user affective state inference process: a method for calculating confidence levels for categorical (CAT) emotion probabilities; a method for transforming CAT-type emotion into the valence, arousal, and dominance (VAD) format; and a multimodal fusion method used to infer the user's affective state.

Since integration is achieved via MQTT messages, these modules are not restricted to any specific programming language, software library, sensing technology, or execution environment, and can run on a machine separate from the other framework components. From this perspective, an input (a source) is identified by a unique source identifier (source ID) and must provide its output using one of the supported representations: categorical emotion probabilities (CAT) or Valence-Arousal-Dominance (VAD).

3.4.1.2 Affective Perception Window

Allowing multimodal input modules to continuously produce and publish observations from the beginning to the end of an interaction would not only consume significant computational resources but could also fail to provide a clear correspondence between the multimodal information and a specific moment within that interaction. To coordinate the timing of affective information capture, the framework defines an Affective Perception Window (APW), essentially a time window dedicated to collecting multimodal inputs. To implement this APW, this work extended the EvaML language by adding a `<perception>` element with two attributes: a mandatory `action` attribute and an optional `sources` attribute. The `action` attribute can take the values "START" or "END," while the `sources` attribute must contain a list of source IDs to be used during the window. The time boundaries ("START" and "END") and the participating sources are explicitly defined by the script author based on interaction requirements. A window is started in an EvaML script using the command `<perception action="START" sources="[FER, TER, ...]" />`, where the attribute `sources` specifies the identifiers of the input module that are to participate in the window. The end of the window is defined by the command `<perception action="END" />`.

When SENSEI initiates an APW, a message containing the "START" action and the defined source IDs (`source` attribute) is published to the `/PERCEPTION` subtopic. All input modules must be subscribed to this topic and check whether their source IDs are included in the message. Upon identifying its ID, a module activates its specific acquisition and processing procedure for that modality, while modules that do not identify their source ID remain inactive. Data acquisition proceeds independently in each activated module until SENSEI issues the command to end the window by publishing a message to the `/PERCEPTION` subtopic containing the "END" action.

Upon receiving the "END" command, each active input module terminates its acquisition, generating a single final result that represents (summarizes) the information obtained during the APW. For example, if an emotion recognition module classified 30 video frames during the window, each with a set of probabilities for every emotion, the module must send a single overall classification for that group of frames. The specific sensing technology, acquisition procedure, and processing technique employed during the window are left to the implementation of each module.

The results from each module active during the window are then published to the `/PULSE/INPUT` subtopic, along with its source ID and representation type. CAT re-

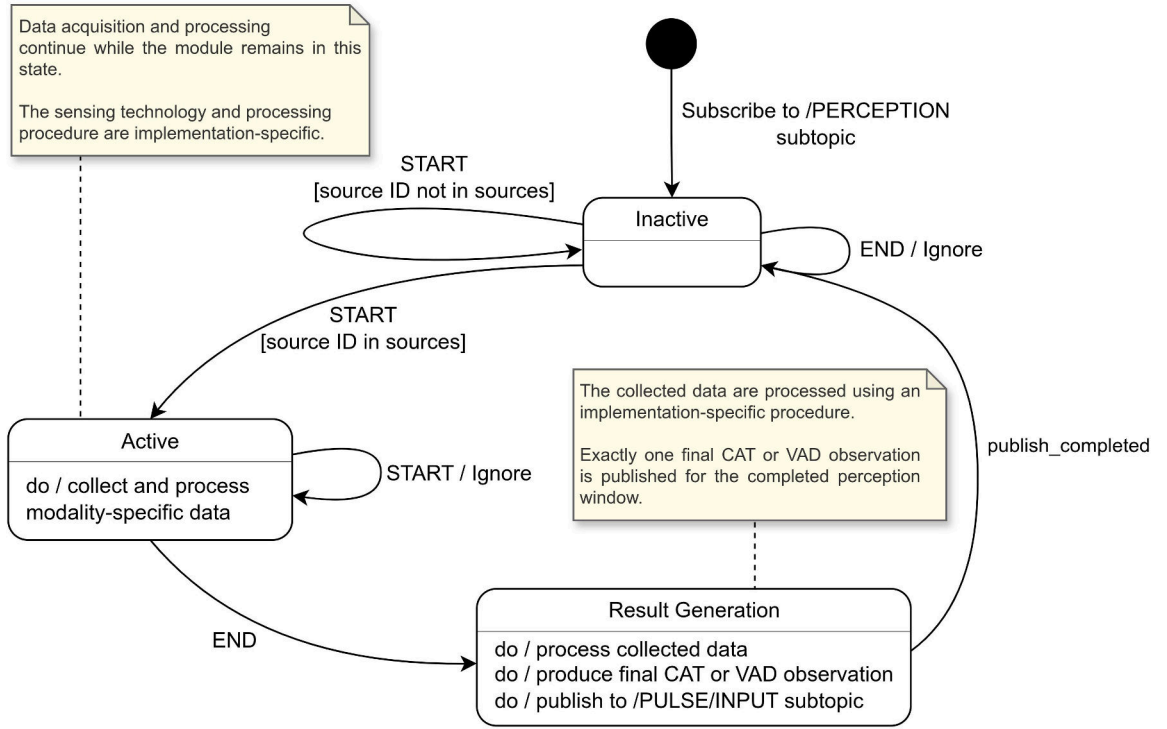


Figure 13: Finite-state machine (FSM) model representing the lifecycle transitions of multimodal input modules.

sults must contain categorical emotions and their respective probabilities, while VAD results must contain values for valence, arousal, and dominance, as well as a confidence value between 0 and 1. This constitutes the communication contract required for an external multimodal input module to participate in the affective state estimation process. The formal communication protocol and state transitions that guide the interaction between the multimodal input modules, SENSEI and PULSE, are modeled as a Finite State Machine, as illustrated in Figure 13.

PULSE also receives and processes "START" action messages published on the /PERCEPTION subtopic, but its role in the protocol differs from that of the input modules. Upon receiving a "START" message, PULSE registers the source IDs specified in the [sources](#) attribute as the set of expected observations for the current APW. As each result sent by a source arrives via the /PULSE/INPUT subtopic, its source ID is removed from the set of pending sources, and the received result (observation) is stored for subsequent processing. Consequently, PULSE does not rely on the "END" action message to determine when the multimodal information is complete. The processing flow is triggered when the set of pending sources becomes empty, indicating that a result has been received from each of the sources defined at the start of the window.

This mechanism decouples the temporal coordination of multimodal perception from

the implementation of individual input modules. The interaction script determines when user affective states are required and which sources (multimodal inputs) must participate, while each external module independently determines how its information is acquired and processed. PULSE, in turn, operates only on the resulting representations (CAT or VAD), allowing for the incorporation of new multimodal sources without the need to add modality-specific processing mechanisms to the system architecture.

Figure 14 shows an example of an APW using two multimodal inputs: one that recognizes emotions from text (TER), specifically, sentences, and another that recognizes emotions from facial expressions (FER). As shown in the upper part of the figure, a timeline represents the duration of the APW, starting with the message containing the "START" action (on the far left) and ending with the message containing the "END" action (on the far right). These positions on the time axis are established as instants t_0 and t_1 , respectively. Data collection (observations) from the two inputs occurs in parallel and independently, as each input runs in its own process, potentially even on different machines. For TER, the input module generates a classification for each sentence in the interaction at successive time points: $t_0 + \Delta t_1$, $t_0 + \Delta t_2$, up to $t_0 + \Delta t_n$. As shown, the emotion recognition model employs a categorical (CAT) representation, outputting probabilities for each of the seven classified emotions. The facial recognition module similarly outputs data in the CAT format, covering seven emotions. Classifications are performed at a frame-based frequency (frame k_1 , frame k_2 , ..., frame k_m). Upon completion of the window (at instant t_1), the inputs employ a method to generate a classification representing (or summarizing) the emotions identified during the APW's duration. These results (observations), one per input are published to the /PULSE/INPUT subtopic and transmitted to PULSE, which stores them in an observation collection. Upon receiving results from both inputs, PULSE identifies that the APW has closed and proceeds to stage 3 of its execution pipeline, calculating the confidence levels for the probabilities associated with each observation. In stage 4, PULSE converts these observations from CAT format to VAD format. It then reaches stage 5, where these multimodal observations are fused to generate a VAD vector representing the user's affective state during the APW. Finally, this VAD vector is published to the /USER_AFFECTIVE_STATE subtopic, reaching the components interested in this information: SENSEI and ROSE. Figure 15 complements this explanation with a more formal representation using a UML sequence diagram, detailing the message exchange and coordination among SENSEI, the input modules, and PULSE.

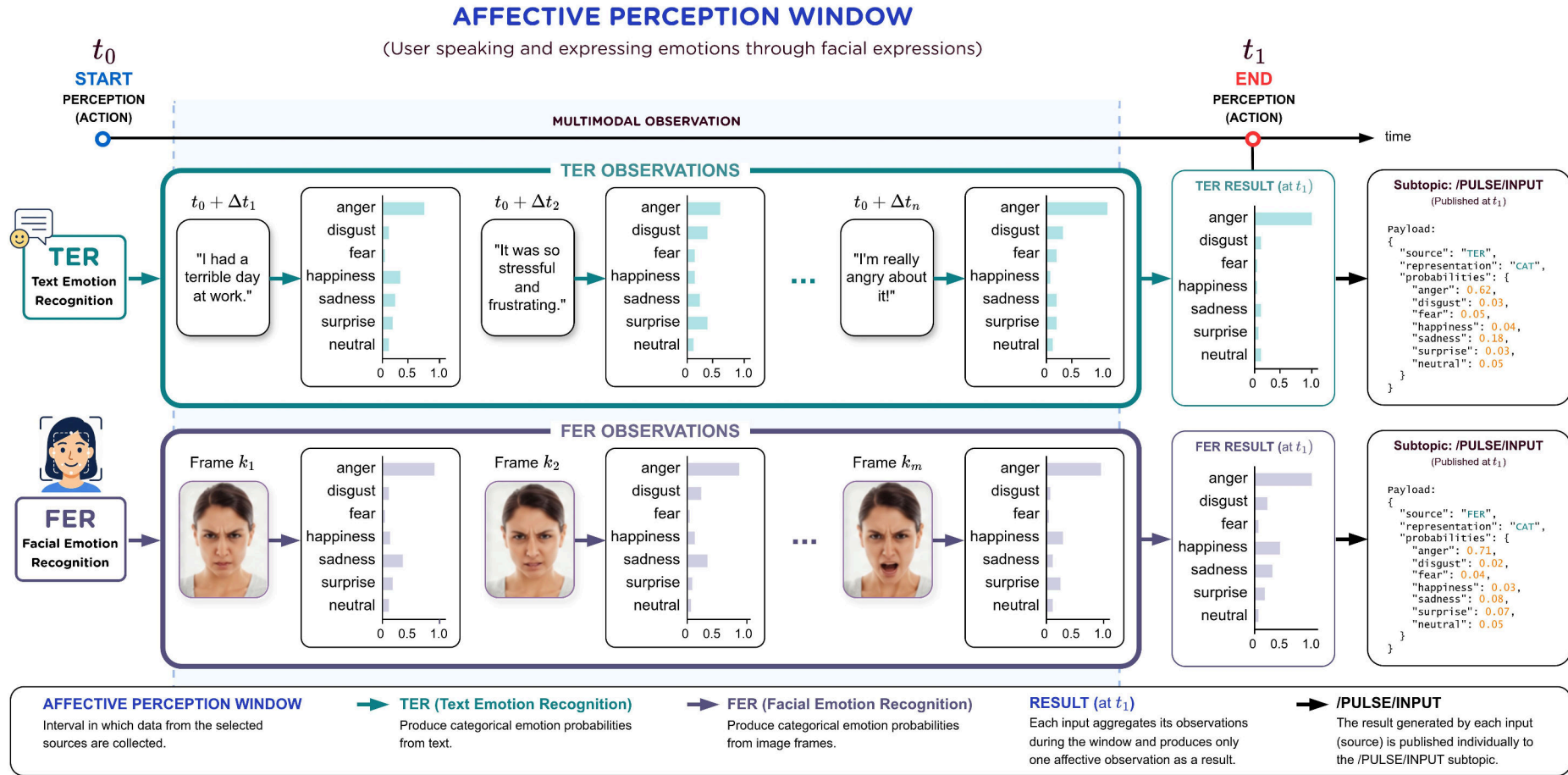


Figure 14: Multimodal input modules (TER and FER) independently collecting and processing modality-specific data and each producing a single observation for the PULSE at the end of the Affective Perception Window. The probability values shown in the figure's bar charts are for illustration purposes only and do not represent plausible values.

3.4.1.3 PULSE’s Extensible Affective State Estimation Methods

As previously mentioned, the processing pipeline defined by PULSE utilizes three stages, confidence estimation, CAT-to-VAD conversion, and multimodal fusion, whose specific methods may vary according to application requirements. Instead of providing specific algorithms for these stages, PULSE defines them as extension points, allowing users to provide their own processing methods. This flexibility is implemented through the abstract classes shown in Figure 16. `ConfidenceMethod`, `CatToVadMethod`, and `FusionMethod` define the contracts required by PULSE for confidence estimation, CAT-to-VAD conversion, and multimodal fusion, respectively. PULSE operates based on these abstractions and does not provide a specific implementation for any of the three processing stages, this allows different approaches to be incorporated by implementing the corresponding abstract class and providing the resulting object to PULSE.

This section introduced the first component of the framework’s Affective Layer: PULSE. It is responsible for estimating the user’s affective state based on multimodal information. The component acquires data from multimodal inputs via a communication protocol that handles information transfer and synchronization messages among the inputs, SENSEI, and PULSE, while remaining independent of the technologies and processing approaches employed by external input modules. The Affective Perception Window was also presented; it coordinates the collection of observations from sources selected for each interaction, while implementing extension points allow for the independent adoption of different methods for confidence estimation, CAT-to-VAD conversion, and multimodal fusion. The next section describes the affective component responsible for modeling the robot’s affective state, as well as an empathic transformation model that alters the robot’s affective vector based on the interaction with the user’s affective vector. These mechanisms provide the architectural approach proposed in this work to address Research Question 2, regarding how can a social robot estimate the user’s affective state.

3.4.2 ROSE – Robot Sentiment Engine

ROSE is the component of the Affective Layer responsible for dynamically maintaining and updating the robot’s affective state during an interaction. While PULSE estimates the user’s affective state based on perceptual information, ROSE uses this state as input to influence the robot’s own affective state. Both states are represented in the three-dimensional Valence-Arousal-Dominance (VAD) space.

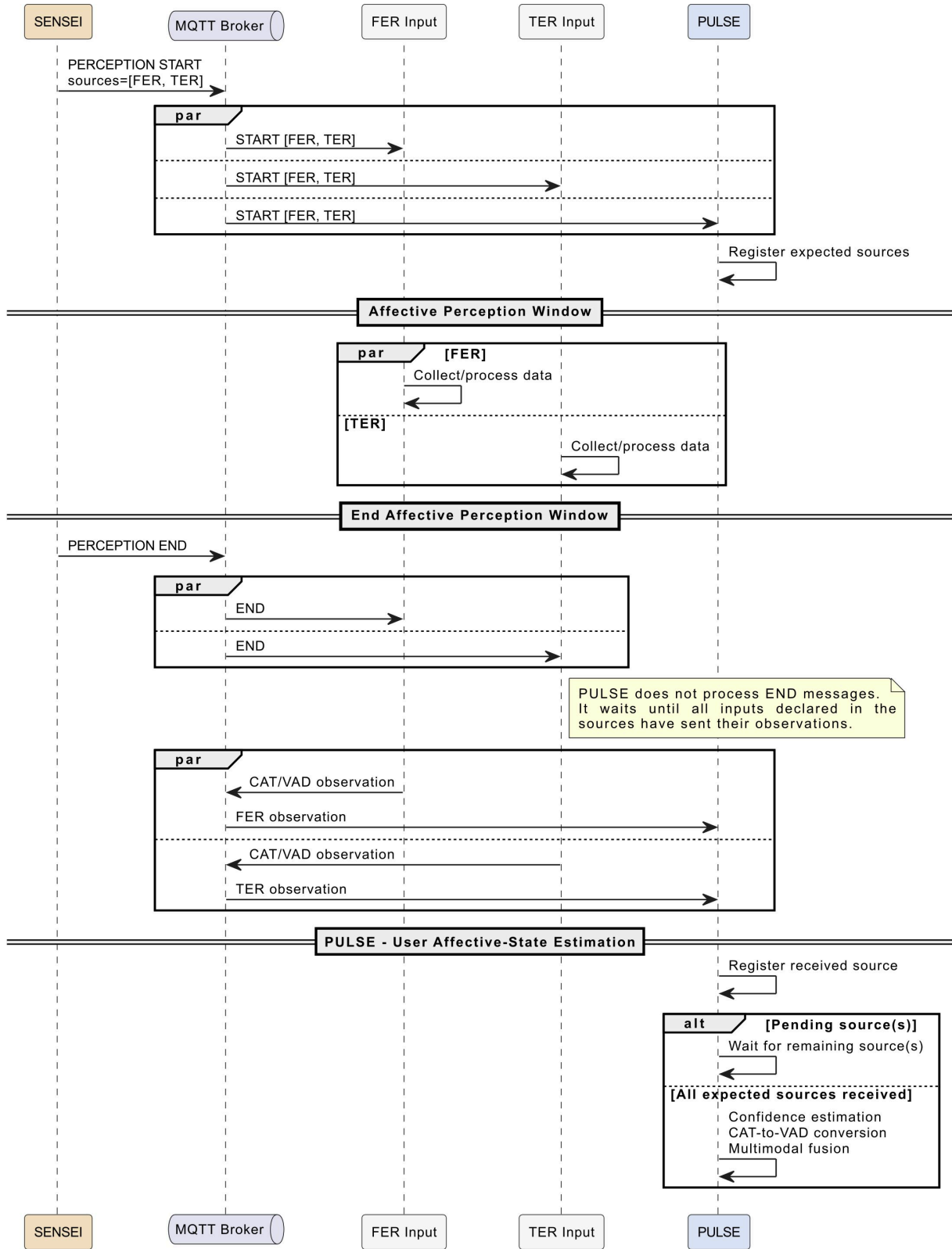


Figure 15: Sequence diagram illustrating the message-based coordination between SENSEI, affective input modules, and PULSE during an Affective Perception Window.

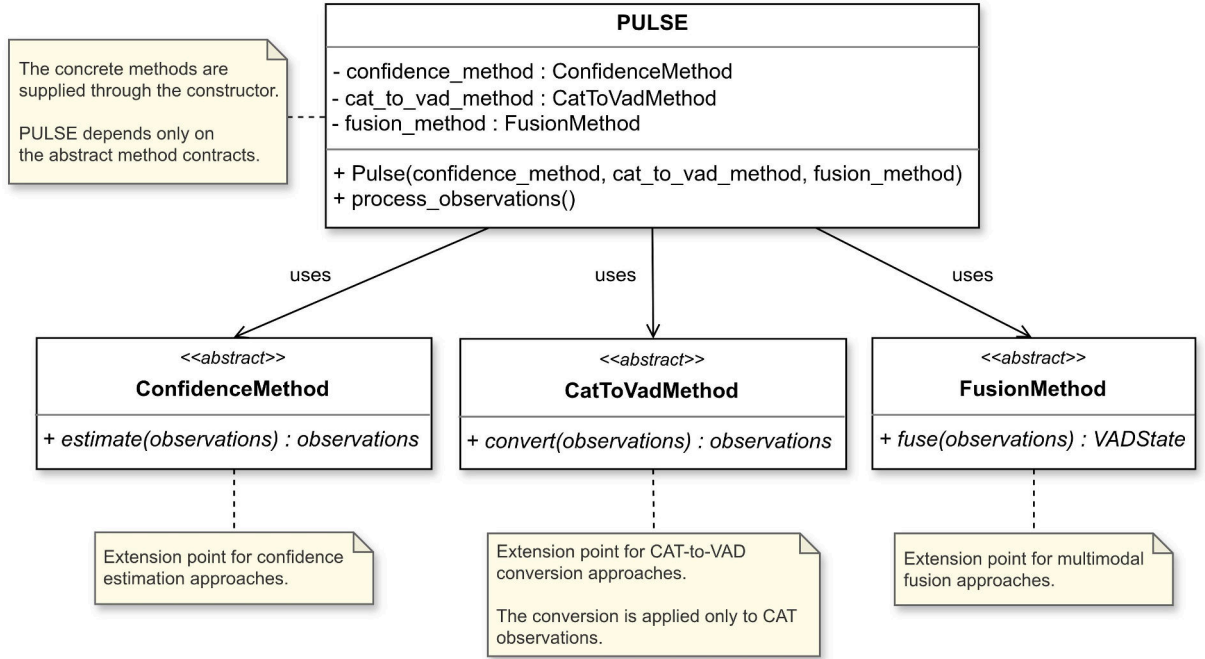


Figure 16: Class diagram of the PULSE extension points for confidence estimation, CAT-to-VAD conversion, and multimodal fusion.

In ROSE, the robot's affective state is denoted by the vector $\mathbf{s}_r = [v_r, a_r, d_r]$, while the user's affective state, received from PULSE, is represented by the vector $\mathbf{s}_u = [v_u, a_u, d_u]$. Thus, ROSE maintains the robot's own affective state and operates on it dynamically. During an interaction with the user, rather than directly assigning the user's state to the robot (thereby mimicking the user's affective state), ROSE applies a mathematical function, generically termed Empathic Transformation in this thesis, that determines a transformed affective state vector, $(\mathbf{s}'_r)$ for the robot. In addition to the user's and robot's affective vectors, this transformation may utilize other parameters defined internally within ROSE. These parameters were established to provide the robot with a specific affective profile, allowing the robot's affective states to be modulated through their configuration. These values define a characteristic of the robot designated as the Robot Affective Profile. The parameters that form this characteristic are: empathy factor e , the base mood $\mathbf{b}$, the emotional decay factor d , and the decay delay ℓ .

The empathy factor (e) is a scalar value in the interval $[0, 1]$ that can be used to control the degree to which the user's affective state vector $\mathbf{s}_u$ influences the transformation of the robot's affective state. The base mood ($\mathbf{b}$) is a VAD-type vector whose components can have values in the interval $[-1, 1]$ intended to represent a persistent affective tendency of the robot. In other words, following a change in its affective state, the robot tends to return to the position defined by $\mathbf{b}$, which effectively defines an affective characteristic of

the robot. This return can be modulated by a parameter d , the emotional decay factor. Its function is to allow the speed of this return to the base mood to be parameterized by this variable, which is defined in the robot's affective profile. Its values are of the scalar type within the range $[0, 1]$. The final profile parameter relates to time. As the name implies, the decay delay (ℓ) determines a pause duration, in seconds, during which the robot's affective state, following the empathic transformation, begins to return to the base mood.

The concept of the Robot Affective Profile was defined in the EvaML language through its complete definition in the XML Schemas folder and the corresponding Python module, both located within the `robot_package` folder. As a configuration command, it is designed for use within the `<settings>` section. The EvaML command is named `<robotAffectiveProfile>` and includes the attributes `empathy`, `decay`, `decayDelay`, and `baseMood`. Additionally, there is a `profile` attribute, which selects from a list of predefined profiles established in the command's Python module. Each profile comes with a set of predefined values for the variables that define it, examples include: NEUTRAL, EMPATHETIC, THERAPIST, OPTIMISTIC and MELANCHOLIC. Each of these profiles utilizes values intended, in a purely exploratory manner, to characterize an affective state corresponding to the profile's name. When executing a script, SENSEI reads the attributes of the `<robotAffectiveProfile>` command and sends them to ROSE via an MQTT message published to the specific subtopic named `/ROBOT_AFFECTIVE_PROFILE`.

Figure 17 presents an overview of the ROSE architecture, including some of the elements introduced so far: the affective state vectors of the user and the robot, the VAD space, and the parameters used to define the Robot Affective Profile in ROSE. Three methods are also shown in the figure. One is the Robot Affective State Update, implemented in ROSE, which controls the transition of the robot's affective states over time. The other two methods, highlighted by rectangles with thicker lines, are abstract methods that define a contract for implementing two mathematical functions: the first one transforms the robot's affective state during interaction based on the user's affective state, and the second one returns the robot's affective state to the base mood, a position in the emotional space representing an affective tendency defined in its affective profile.

3.4.2.1 Dynamic Robot Affective State Model

The primary responsibility of ROSE is to maintain the robot's affective state as a dynamic state that evolves over time during interaction with the user. This dynamic state is con-

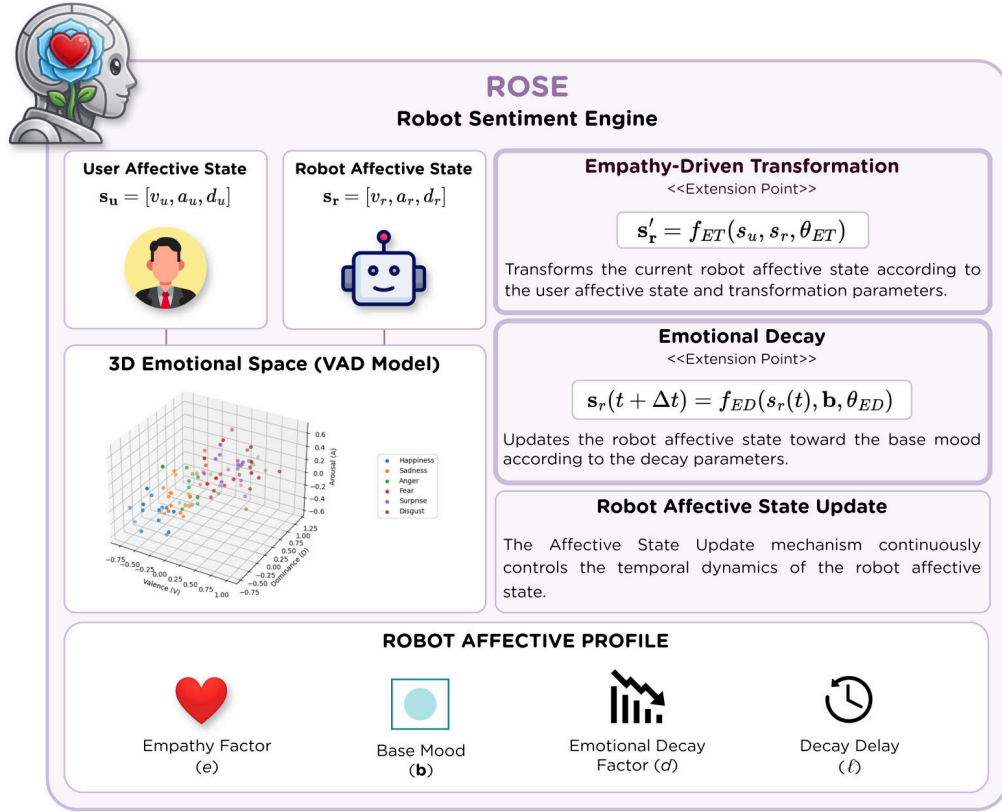


Figure 17: Overview of the ROSE architecture, including the robot's affective profile parameters and its extension points.

trolled by the `robot_affective_state_update()` method, which continuously manages the robot's affective state based on affective information received from PULSE, and the configuration defined in the Robot's Affective Profile.

When a new user affective state is received from PULSE, the function designated as Empathic Transformation, determines a new resulting robot affective state, $\mathbf{s}'_r$. From this point on, the method responsible for updating the robot's affective state comes into play and immediately changes the robot's current affective state $\mathbf{s}_r$ to $\mathbf{s}'_r$. Once the target state is reached, it is maintained for the duration specified by the decay delay ℓ . During this period, no return to the base mood occurs. If a new affective state of the user is sent by PULSE during this period, a new empathetic transformation is triggered, establishing a new $\mathbf{s}'_r$ and restarting the time count established by the decay delay value (ℓ).

Upon completion of the decay delay ℓ , the robot affective state update initiates the return of $\mathbf{s}_r$ toward the base mood $\mathbf{b}$. This return occurs progressively rather than as an instantaneous transition, with the emotional decay factor d serving as the parameter regulating this process. The robot's affective state is successively updated toward $\mathbf{b}$ until the target affective state is reached. If a new affective state of the user is sent by PULSE

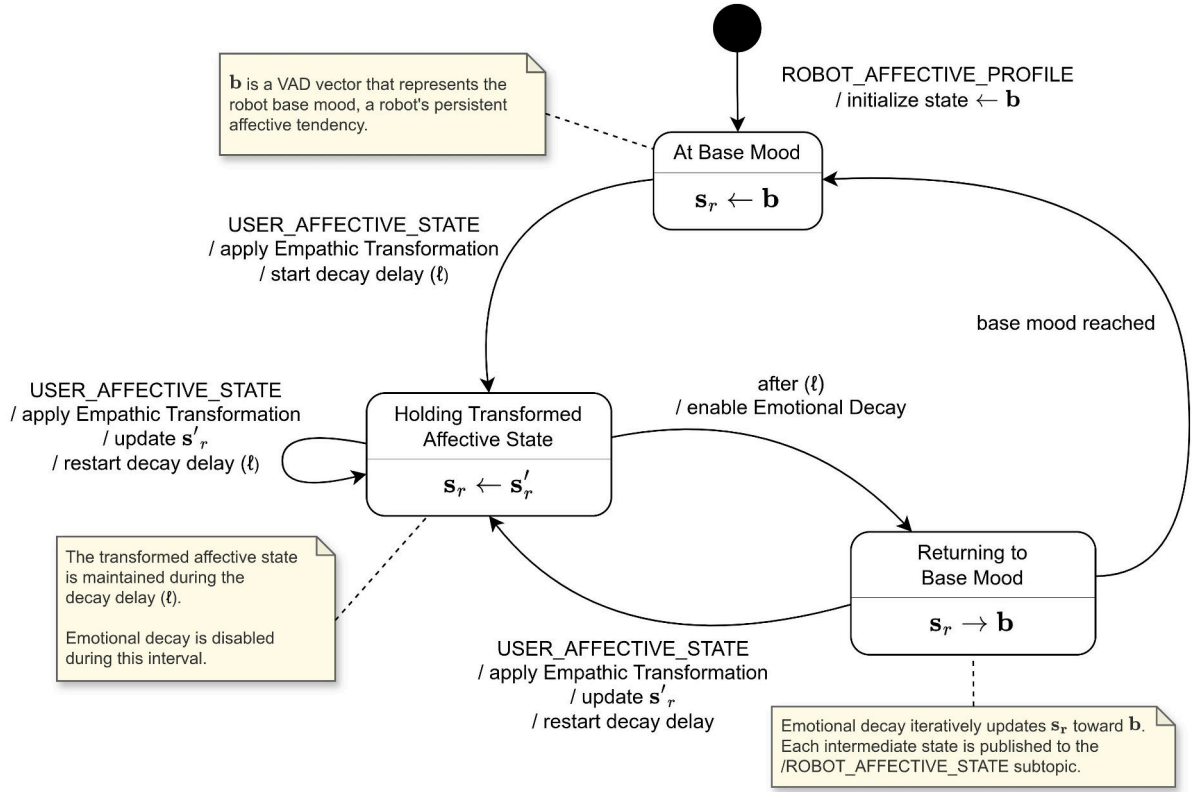


Figure 18: UML state machine diagram representing the dynamics of the robot's affective state in ROSE.

during this period, just as in the period defined by ℓ , a new empathic transformation is triggered, establishing a new $\mathbf{s}'_r$ and restarting the time count established by the decay delay value (ℓ).

Figure 18 represents this behavior as a UML state machine. The "At Base Mood" state represents the condition of the robot's persistent affective tendency, in which $\mathbf{s}_r = \mathbf{b}$. The "Holding Transformed Affective State" represents the condition in which $\mathbf{s}_r = \mathbf{s}'_r$ and this state is maintained for a duration of ℓ . Finally, the "Returning to Base Mood" state represents the progressive evolution of $\mathbf{s}_r$ toward $\mathbf{b}$. Transitions triggered by new user affective states illustrate that affective dynamics can be restarted at any point during the interaction.

3.4.2.2 ROSE's Extensible Affective State Transformations

The ROSE Affective State Dynamic Model defines the temporal behavior of the robot affective state, including its transition to a transformed state, its persistence during the decay delay, and its return toward the base mood. However, ROSE does not define how the affective states involved in these transitions must be mathematically computed.

Instead, two operations of this model, Empathic Transformation and Emotional Decay, are defined as extension points, allowing different affective transformation methods to be incorporated according to the requirements of the interaction.

The Empathic Transformation determines a transformed robot affective state vector $\mathbf{s}'_r$ considering the user affective state vector $\mathbf{s}_u$, the current robot affective state vector $\mathbf{s}_r$, and a vector of transformation-specific parameters. This operation can be generically represented as $\mathbf{s}'_r = f_{ET}(\mathbf{s}_u, \mathbf{s}_r, \boldsymbol{\theta}_{ET})$, where f_{ET} denotes an Empathic Transformation function and $\boldsymbol{\theta}_{ET}$ represents the vector of parameters required by its implementation. This abstraction allows different approaches to determine how the user's affective state influences the robot's affective state without modifying the dynamic model maintained by ROSE.

The Emotional Decay constitutes the second extension point and determines the successive affective states during the robot's return toward its base mood $\mathbf{b}$. This operation can be generically represented as $\mathbf{s}_r(t + \Delta t) = f_{ED}(\mathbf{s}_r(t), \mathbf{b}, \boldsymbol{\theta}_{ED})$, where f_{ED} denotes an Emotional Decay function and $\boldsymbol{\theta}_{ED}$ represents the vector of parameters required by its implementation. The method `robot_affective_state_update()`, after the decay delay (ℓ) has elapsed, repeatedly applies this transformation, generating successive values of $\mathbf{s}_r$ toward the base mood. The temporal condition that determines when this return begins therefore remains part of the ROSE dynamic model, whereas the computation of each successive affective state is delegated to the Emotional Decay extension point.

Figure 19 presents the UML class diagram used to implement these extension points. The abstract classes `EmpathicTransformationMethod` and `EmotionalDecayMethod` define the interfaces for computing the transformed robot affective state and the successive affective states toward the base mood, respectively. Concrete implementations of these abstractions are provided to ROSE through *dependency injection*, a pattern also used in SENSEI allowing alternative transformation and decay methods to be employed without modifying the affective state update mechanism, which remains a non-extensible part of ROSE.

This section presented the second component of the Affective Layer of the framework. ROSE proposes a dynamic model for maintaining and updating the robot's affective state based on the user's affective state, the application of an empathy-driven transformation function, and the parameters defined in the configured Robot Affective Profile. Its extensible affective state transformations allow different mathematical formulations to be implemented and adopted while preserving the temporal dynamics internally defined by

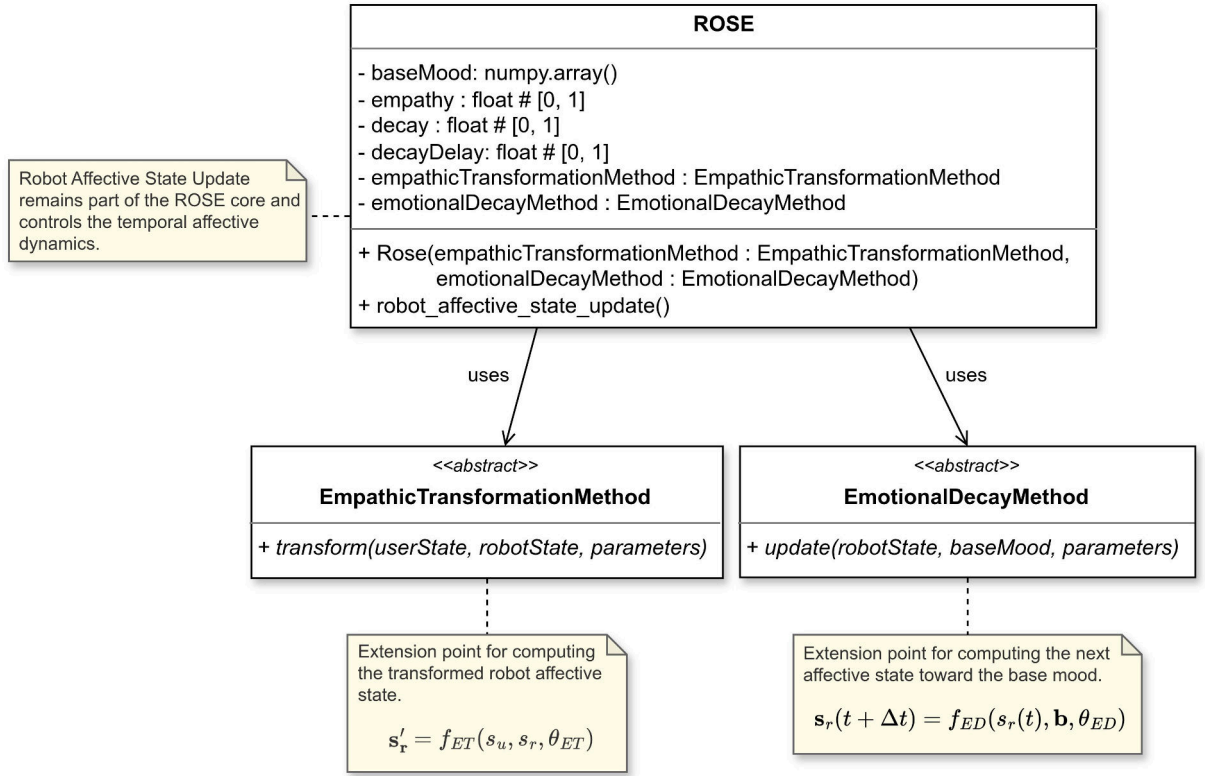


Figure 19: UML class diagram of ROSE showing the extension points for Empathic Transformation and Emotional Decay.

the component. To complete the affective interaction loop, the robot's resulting affective state must be mapped into multimodal behavioral specifications, which may include elements such as tone of voice, gaze expressions, LED colors, postures, and movements, producing adaptive affective behavior capable of making interactions with the user more natural and engaging. The AEB component, presented in the following section, is proposed to address this stage.

3.4.3 Automatic Empathic Behavior – AEB

The Automatic Empathic Behavior (AEB) component was initially designed to generate automatic robot behavior in contexts requiring an empathetic response. However, in the architecture proposed in this thesis, the AEB assumes a broader role and is not restricted to generating empathetic behavior. It has become responsible for automatically mapping the robot's affective state to multimodal behavioral specifications. The name AEB has been retained in this thesis to ensure consistency with earlier versions of the component, which were originally defined and applied in Sections 4.2, 6, and 6.3.

While ROSE maintains and updates the robot's affective state during the interaction,

the AEB acts upon this resulting state and converts it into a structured representation, which may include elements of verbal and non-verbal expression, with the aim of generating affective behavior that can be executed by the framework’s Execution Layer.

Figure 20 illustrates the component’s internal architecture. Processing begins when the AEB receives the robot’s current affective state through the vector $\mathbf{s}_r$, represented in VAD space. The Affective State Processing mechanism coordinates the processing of this state and invokes the Empathic Behavior Mapping method. This mapping constitutes the component’s primary extension point and determines how the robot’s affective state is mapped into multimodal behavioral specifications. The resulting parameters are represented as a key-value structure and can describe various forms of robot expression, such as facial expressions, postures, LED colors, sounds, movements, speech characteristics, gaze direction, or lighting effects. The specific elements contained in this structure are not fixed by the AEB and may vary according to the expressive capabilities and requirements of the target robotic platform. Therefore, the AEB does not execute the behavior itself but generates these parameters, which can subsequently be interpreted by the framework component responsible for behavior realization.

The Behavior Parameter Publishing mechanism receives the structured parameters generated by the mapping method and makes them available to the Execution Layer component, SENSEI. This separation allows the AEB core to remain independent of the specific strategy adopted to map and generate behavior parameters (multimodal expressions), as well as the specific physical mechanisms used by the robot to express them.

3.4.3.1 AEB’s Extensible Empathic Behavior Mapping

The AEB does not define a specific algorithm for mapping the robot’s affective state into behavior parameters. Instead, the Empathic Behavior Mapping is defined as an extension point, allowing different mapping strategies to be incorporated. This operation can be generically represented as $p = f_{EB}(\mathbf{s}_r, \boldsymbol{\theta}_{EB})$, where f_{EB} represents the Empathic Behavior Mapping function, $\mathbf{s}_r$ represents the robot’s current affective state, $\boldsymbol{\theta}_{EB}$ represents optional implementation-specific parameters, and p represents the resulting structured set of robot behavior parameters.

A specific implementation of Empathic Behavior Mapping may introduce intermediate affective representations, reference points, discrete emotional levels, rules, or distance measures, while another implementation may map the VAD values directly to behavioral parameters. The mapping function may also be implemented to generate behavior de-

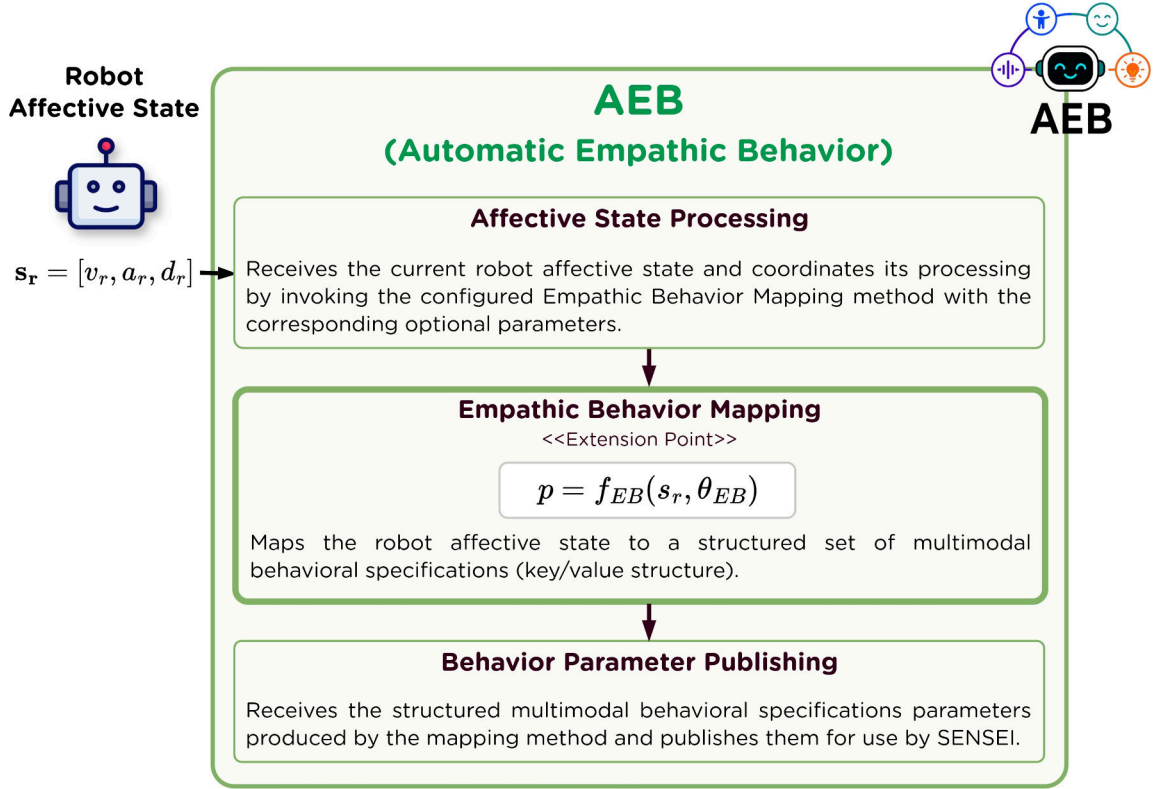


Figure 20: Overview of the AEB component architecture. The Empathic Behavior Mapping extension point is highlighted in the figure.

descriptions using Behavior Markup Language (BML) (KOPP et al., 2006; HOLROYD; RICH, 2012; MUSICCO; TRAETTA, 2026). It is an XML-based language for specifying and coordinating multimodal behaviors of embodied conversational agents, including modalities such as speech, gaze, facial expressions, gestures, and body movements

Figure 21 presents the class diagram associated with this extension point. The abstract class `EmpathicBehaviorMappingMethod` defines the `map()` operation used by the AEB to obtain the structured behavior parameters. A concrete implementation of this abstraction must be provided to the AEB component, which relies only on the abstract mapping contract. The `process_robot_affective_state()` method coordinates the use of the chosen mapping method, while `publish_behavior_parameters()` publishes the resulting parameter structure.

This section introduced AEB, the third and final component of the Affective Layer, responsible for automatically mapping the robot's affective state into multimodal behavioral specifications. Its extensible mapping mechanism allows for the adoption of various mapping strategies supporting different interaction requirements and expressive capabilities specific to the target robotic platform. By decoupling the mapping strategy from internal processing and publication mechanisms, the AEB provides an adaptable mecha-

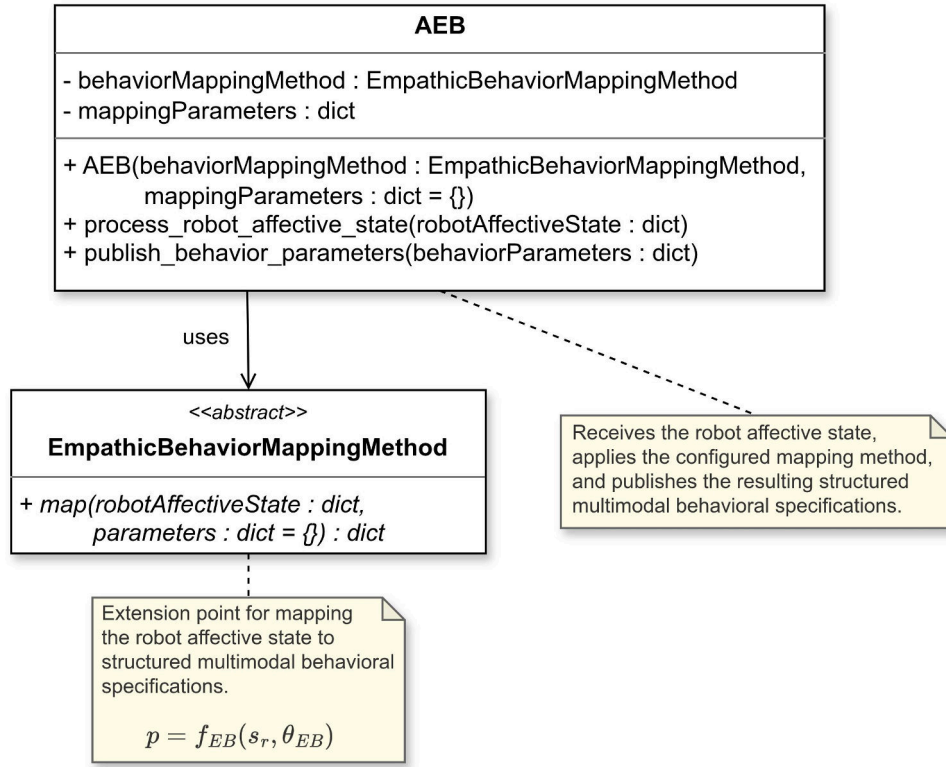


Figure 21: UML class diagram of the AEB component and its extensible Empathic Behavior Mapping.

nism for linking the robot's affective state to its verbal and non-verbal behavior, thereby completing the affective processing stages proposed by the Affective Layer.

3.5 Architectural and Design Patterns

The objective of this section is to summarize the main architectural and design patterns adopted in the proposed framework, describing the use of each, the benefits associated with these choices, and the references that provide their theoretical foundation, based on the established literature in the areas of software architecture and design patterns.

The design of the proposed framework was guided by architectural decisions aimed at promoting modularity, extensibility, and maintainability. These principles are reflected in its layers, particularly through the use of abstractions, interfaces, and extension points that allow for the incorporation of alternative implementations without altering the overall architecture. However, these design decisions are most explicitly represented in SENSEI, where multiple architectural and object-oriented design patterns are combined to enable communication independence, dynamic module loading, and the extensibility of EvaML commands. Therefore, SENSEI is used in this section as the primary implementation

Table 3: The main architectural and software design approaches adopted in the proposed framework, their implementation, and their respective objectives.

Pattern/Approach	Main implementation	Purpose
Strategy	CommunicatorInterface and its concrete implementations	Supports multiple communication mechanisms through a uniform interface.
Simple Factory	CommunicatorFactory	Creates the communicator selected by the <i>commMode</i> attribute.
Null Object	NullCommunicator	Provides a valid communication object when no message exchange is required.
Dependency Injection	ModuleLoader	Supplies the selected communicator to each CommandHandler during initialization.
Microkernel (Plugin-Based) Architecture	ModuleLoader and <i>robot_package</i>	Supports dynamic loading and runtime extensibility of EvaML modules.
Publish-Subscribe	MQTT infrastructure	Decouples distributed message publishers and subscribers.

reference to illustrate the architectural and design patterns adopted in the framework. Table 3 summarizes these patterns, their main implementations, and their purposes.

Although based on the publish-subscribe paradigm, the SENSEI architecture allows other protocols to be used in the implementation of new commands and functionalities for EvaML. To achieve this goal, the communication mechanism adopted in SENSEI combines the Strategy (GAMMA et al., 1994), Simple Factory (FREEMAN; ROBSON, 2020), and Null Object (WOOLF, 1998) patterns to decouple communication mechanisms from the semantics of EvaML elements. Command handlers interact exclusively with the CommunicatorInterface, while the CommunicatorFactory selects the appropriate implementation according to the *commMode* attribute defined for each EvaML element. Commands that do not require external communication receive a NullCommunicator, allowing all handlers to operate through the same interface without introducing conditional communication logic. Figure 22 illustrates these relationships.

During module loading, each CommandHandler is configured before becoming available for script execution. The ModuleLoader requests the appropriate communicator from the CommunicatorFactory. The resulting CommunicatorInterface instance is injected into the handler constructor, and the configured handler is stored in the modules table (`tab_modules`). This constructor-based dependency injection (FOWLER, 2004) separates object creation from object usage, reducing coupling between command handlers and communication mechanisms while simplifying maintenance and future extensions. Figure 23 presents a small sequence diagram showing how dependency injection occurs and which classes, objects and interfaces are involved in the process.

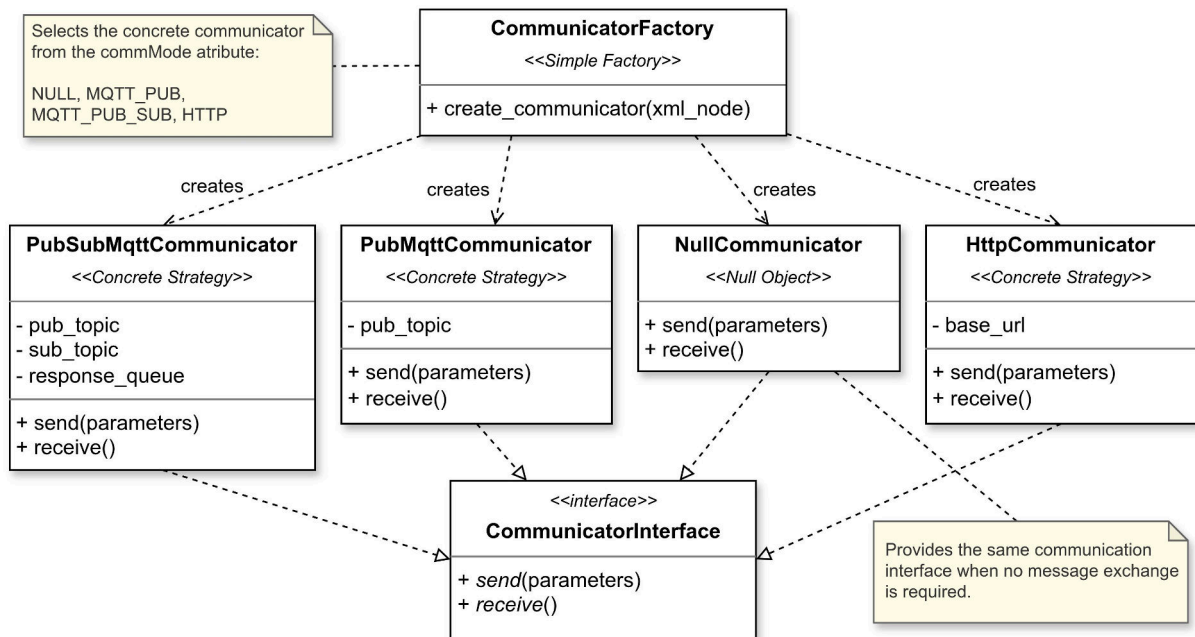


Figure 22: UML class diagram of the communication architecture in SENSEI. The **CommunicatorFactory** selects the communication strategy defined by the *commMode* attribute, while all communicators implement the common **CommunicatorInterface**.

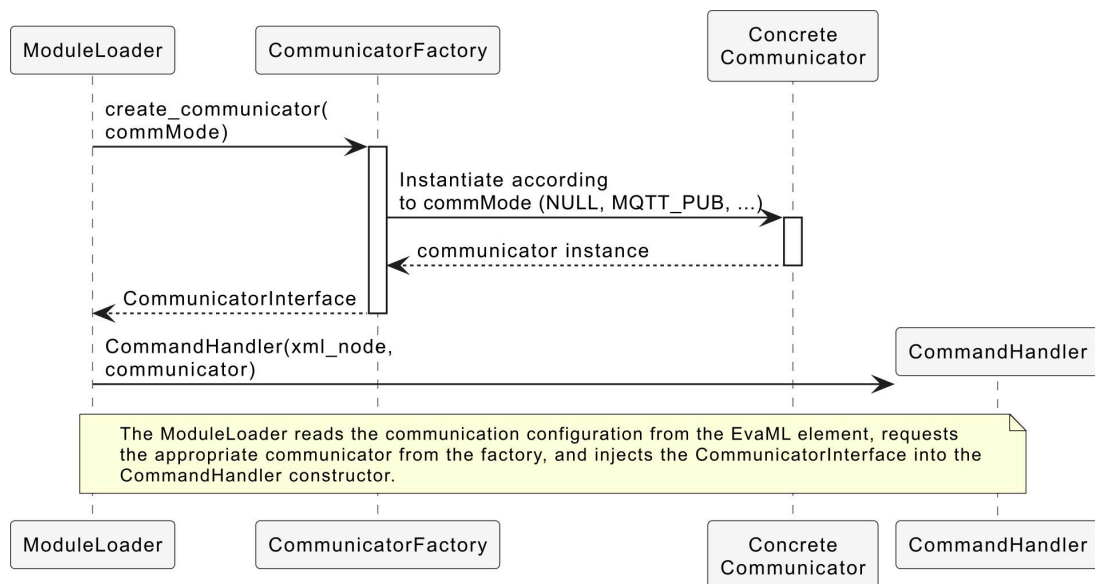


Figure 23: UML sequence diagram illustrating constructor-based dependency injection in SENSEI. Communication objects are created by the **CommunicatorFactory** and injected into **CommandHandler** instances during module loading.

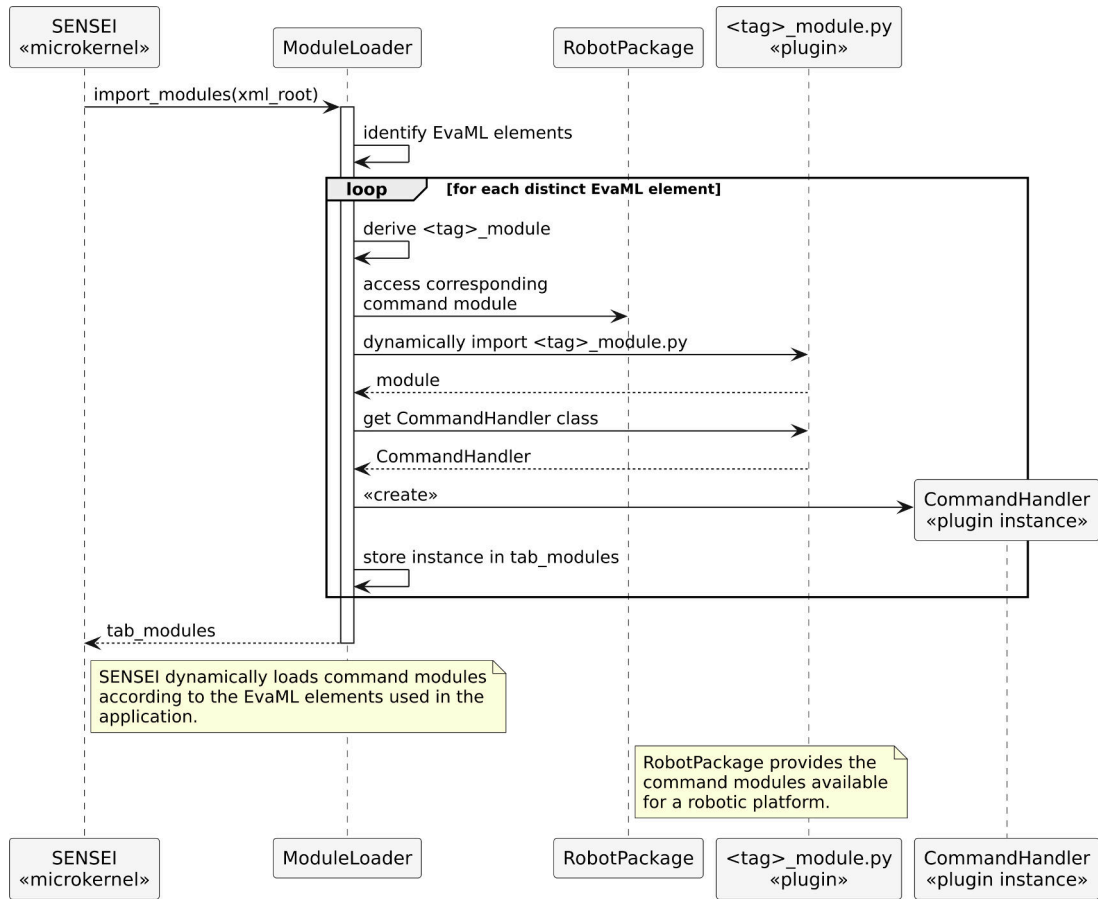


Figure 24: UML sequence diagram illustrating the dynamic loading of command modules in SENSEI using the Microkernel/Plugin architectural approach.

Modular language extensibility is achieved through a Microkernel/Plugin architecture pattern (BUSCHMANN et al., 1996) implemented by the ModuleLoader and the `robot_package` directory. Rather than relying on a fixed set of supported commands, the execution engine dynamically identifies the EvaML elements used by the application, imports the corresponding Python modules according to the established naming convention, retrieves their `CommandHandler` implementations, and registers them for subsequent execution. Consequently, new EvaML elements can be incorporated without modifying the execution engine, provided that they follow the expected module organization and programming interface. Figure 24 shows this architectural approach being employed in SENSEI during the dynamic loading of the Python command modules corresponding to the elements contained in the EvaML script to be executed.

As previously mentioned in Section 3.2.1, the framework proposed in this work adopts the Publish-Subscribe (BUSCHMANN et al., 1996) paradigm for message exchange among its components. All distributed interaction involving SENSEI, simulator components, robot-side controls, and affective modules, follows this communication model implemented

over MQTT. It decouples message publishers and subscribers while supporting asynchronous communication and state synchronization across distributed processes. The synchronization of externally generated affective and behavioral states described earlier in this chapter represents an application of this communication model and is illustrated in Figure 7.

3.6 Final Remarks

This chapter presented the architecture of the Social Robotics Framework proposed in this thesis, defining the organization and responsibilities of its three layers and components. The proposed architecture uses architectural and design patterns to achieve modularity, extensibility, and low coupling throughout the framework. Its layered organization separates affective processing, script execution, and robot abstractions according to their respective responsibilities, while the Publish-Subscribe messaging pattern enables distributed components to exchange information through MQTT without direct dependencies between message publishers and subscribers.

The Affective Layer comprises the PULSE, ROSE, and AEB components, which handle, respectively, the estimation of the user's affective state, the dynamic modeling of the robot's affective state, and the automatic mapping of this state into multimodal behavioral specifications. The Execution Layer, represented by SENSEI, is responsible for executing and coordinating interaction scripts, while the Robot Abstraction Layer provides the abstractions needed to manage the robot's capabilities. This organization defines how affective processing, interaction execution, and robot control are integrated within the proposed framework.

The architectures of the PULSE, ROSE, and AEB components were presented with a focus on their extension points, where alternative implementations for some of their responsibilities, affective state estimation (PULSE), affective state transformation (ROSE), and empathetic behavior mapping (AEB), are defined through abstract interfaces. In the Execution Layer, SENSEI adopts a plugin-based extensibility mechanism, allowing command modules associated with EvaML elements to be dynamically discovered and loaded from robot packages.

The architectural definitions presented in this chapter thus provide the basis for instantiating the proposed structure in concrete robotic systems. The following chapter presents these instantiations, describing how the architectural elements defined here may

be implemented according to the characteristics and requirements of different contexts and robotic platforms.

4 Framework Instantiations

The previous chapter presented the components of the proposed Social Robotics Framework, which is divided into Affective Layer, Execution Layer and Robot Abstraction Layer (RAL). In its definitions several extension points were defined. This chapter discusses some instantiations of the framework presenting options that should not be seen as definitive solutions for extension points. This chapter is organized as follows.

Section 4.1 presents the instantiation of the components of the Affective Layer, PULSE, ROSE, and AEB, presenting, at the end, two affective interaction scenarios that demonstrate the integration of this layer with the others, as well as a demonstration video of these scenarios involving a user and a robotic platform that is a complete example of the framework instantiation, including all three layers.

The chapter proceeds by presenting three further instantiations with three different robots: EVA, GRACE, and FRED MX. Specifically, these instantiations refer to the RAL layer, not a complete instantiation of the framework. From this point forward, whenever this text refers to the instantiation of the framework for these three robotic platforms, it will only mean the instantiation of the RAL layer.

Section 4.2 presents the instantiation of the framework represented by the EVA robot. Based on this instantiation and the new EVA control software, the extensions to the robot's non-verbal communication and multimodal interaction capabilities are also presented.

Section 4.3 presents the instantiation of the framework that was made by researchers at ETH Zurich in collaboration with researchers from the University of Zurich, the University of St. Gallen, and Clinic Barmelweid. They created the GRACE robot, which is a robot based on EVA.

Section 4.4 presents FRED MX, which represents another instantiation of the framework proposed in this thesis, developed within the scope of a research project conducted at the Center for Scientific Research and Higher Education of Ensenada (CICESE) in

Mexico. The platform was used in a research project within an educational context.

Finally, Section 4.5 briefly presents two applications: a web application called EvaPlay and a mobile application called EVA Virtual. Both utilize the Execution Layer component, SENSEI, as a script player. These applications exemplify the possibility of building new applications by reusing the framework’s components.

4.1 Instantiation of the Affective Layer

This section presents an instantiation of the Affective Layer. While the architecture defined the responsibilities of PULSE, ROSE, and AEB, it did not specify the algorithms and computational methods to be used at the established extension points. These extension points are implemented through specific implementation choices, providing a concrete example of their operation and integration. These choices are intended to demonstrate one possible implementation of the proposed abstractions rather than prescribe specific implementation methods.

The instantiation comprises the three components of the Affective Layer. For PULSE, concrete methods are provided for confidence estimation, CAT-to-VAD conversion, and multimodal affective state fusion. For ROSE, the instantiation defines the mathematical methods used for the empathy-driven transformation of the user’s affective state and for the temporal decay of the robot’s affective state. Finally, for AEB, a concrete Empathic Behavior Mapping method is implemented to transform the resulting robot affective state into structured behavior parameters. Figure 25 summarizes the relationships between the extension points defined in the framework architecture and the concrete methods adopted in this instantiation.

Following the presentation of each component’s implementation, an affective interaction scenario is used to illustrate the integration of the Affective Layer with the Execution Layer. The scenario is executed using the FRED robot as the robotics platform and employs two different types of Robot Affective Profiles to demonstrate how varying affective profiles can result in different behaviors and affective dynamics. FRED serves only as the platform for this scenario, its design, hardware and software architectures, and implementation details are presented in Chapter 6.

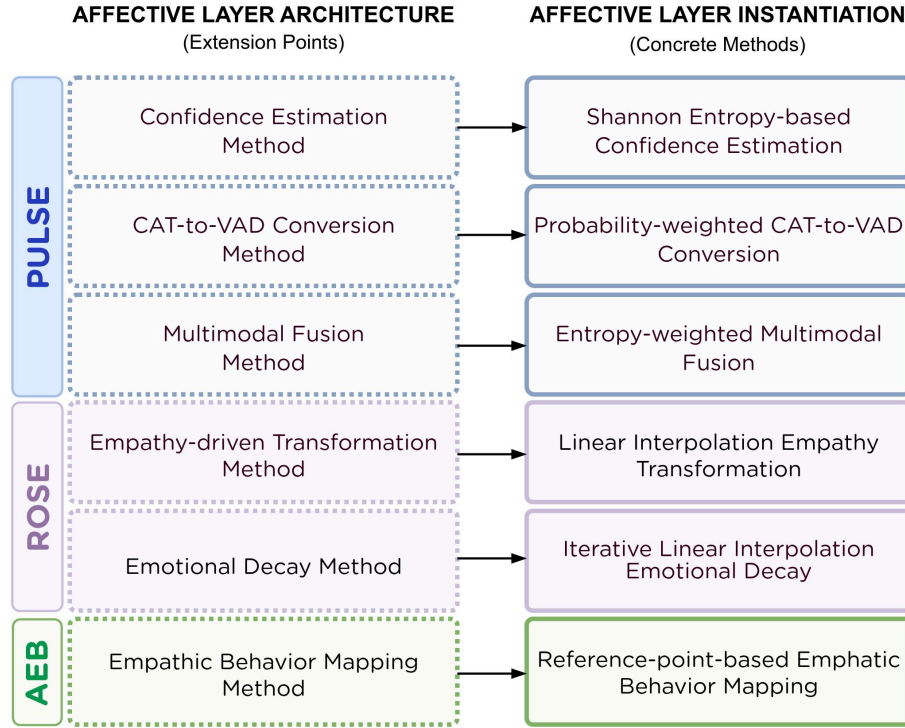


Figure 25: Relationship between the extension points proposed in the Affective Layer architecture and the examples of concrete methods presented for each component.

4.1.1 PULSE Instantiation

This section presents an instantiation of the extension points defined for PULSE. The objective of this instantiation is to demonstrate how concrete computational methods can be incorporated into the component without modifying its architecture or processing interface. The methods adopted here represent one possible implementation of these extension points and do not aim to establish an optimal or preferred solution. Alternative methods may be provided according to the requirements of a specific application.

4.1.1.1 Confidence Estimation Instantiation

To quantify the uncertainty of a model’s predictions and use it as a confidence estimator, Shannon’s Entropy can be applied to the probability distribution of the output (SHANNON, 1948). In this instantiation, the confidence estimation extension point defined by PULSE uses the normalized Shannon’s Entropy. Given a categorical probability distribution $\mathbf{p} = [p_1, \dots, p_K]$ over K affective categories, the normalized entropy is computed as

$$H(\mathbf{p}) = -\frac{\sum_{i=1}^K p_i \log p_i}{\log K}. \quad (4.1)$$

The corresponding confidence value is defined as

$$c = 1 - H(\mathbf{p}), \quad (4.2)$$

where $c \in [0,1]$. Thus, more concentrated probability distributions result in higher confidence values, whereas more uniform distributions result in lower confidence values. This entropy-based approach represents one possible implementation of the confidence estimation extension point and is not prescribed by the framework.

4.1.1.2 CAT-to-VAD Conversion Instantiation

To instantiate the concrete method associated with the CAT-to-VAD conversion, this work employs the probability-weighted mapping used by [Sampaio et al. \(2022\)](#). In this conversion method, each affective category is associated with a reference vector in VAD space. The reference values adopted in the method are those presented in the work of [De Bruyne, De Clercq, and Hoste \(2020\)](#).

For the CAT-to-VAD transformation, authors use a categorical probability distribution $\mathbf{p} = [p_1, \dots, p_K]$, derived from the classification of facial emotions, and the resulting VAD state is computed as

$$\mathbf{s}_{\text{vad}} = \sum_{n=1}^K p_i \mathbf{v}_i, \quad (4.3)$$

where p_i is the probability associated with emotion category i and $\mathbf{v}_i$ is the corresponding VAD vector. This approach represents a possible implementation of the CAT-to-VAD conversion extension point defined in the PULSE architecture and may be replaced by an alternative conversion method. Figure 26 shows the mapping of Ekman's six emotions in a VAD space. The Neutral emotion is located at the center of the graph. The figure caption lists the vector component values corresponding to each emotion.

4.1.1.3 Multimodal Fusion Instantiation

Multimodal classification methods are commonly affected by noise during the classification process. Systems that classify emotions based on facial capture often exhibit low accuracy due to issues regarding face positioning relative to the camera and inadequate

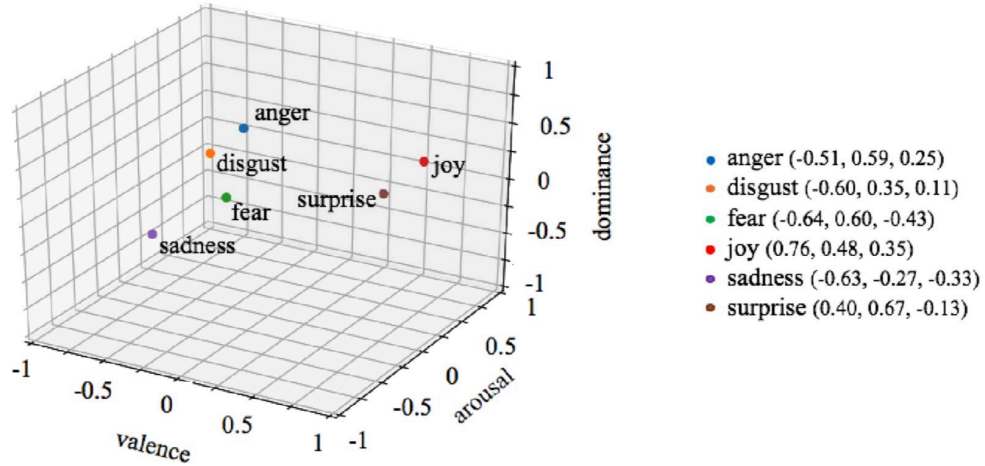


Figure 26: Mapping of Ekman’s six emotions to the VAD space and the coordinates of the anchor points for each. The Neutral emotion corresponds to the coordinate (0, 0, 0) (DE BRUYNE; DE CLERCQ; HOSTE, 2020).

ambient lighting. Systems that classify emotions from voice generally show unsatisfactory performance due to background noise and simultaneous speech from multiple people. In such cases, using a confidence measure within the multimodal fusion process allows for assigning greater importance (weights) to a specific multimodal input over another, thereby ensuring that the most reliable input exerts greater influence on the final classification result. To implement the multimodal fusion extension point, a confidence-weighted average of the VAD states provided by the input modalities, used during the affective perception window, is employed. Given M modalities, the user’s resulting affective state is calculated as

$$\mathbf{s}_u = \frac{\sum_{m=1}^M c_m \mathbf{s}_m}{\sum_{m=1}^M c_m}, \quad (4.4)$$

where $\mathbf{s}_m$ and c_m represent, respectively, the VAD state and confidence associated with modality m . This confidence-weighted average strategy aims to represent a possible implementation of the multimodal fusion extension point and, like other instantiations, can be replaced by another suitable approach.

4.1.2 ROSE Instantiation

The dynamic affective model defined by ROSE is instantiated here using the parameters specified through the Robot Affective Profile, including the empathy factor, emotion decay factor, base mood, and decay delay. These parameters configure how the robot affective state is influenced by the user affective state and how it subsequently evolves toward its base mood. They are not intended to model or reproduce the psychological or biological

processes underlying human emotions, but rather to provide one possible computational realization of the dynamic affective model defined by ROSE.

4.1.2.1 Empathy-Driven Transformation Instantiation

Empathy-Driven Transformation consists of an operation that transforms the vector representing the robot's affective state based on the vector representing the user's affective state. Its instantiation is achieved through linear interpolation. Given the robot state $\mathbf{s}_r$, the current user state $\mathbf{s}_u$, and the empathy factor e defined in the Robot Affective Profile, the updated state is computed as

$$\mathbf{s}'_r = e \cdot \mathbf{s}_u + (1 - e) \cdot \mathbf{s}_r, \quad (4.5)$$

where the Empathic Factor $e \in [0,1]$ serves to determine the influence of the user's affective state on the outcome of the interaction with the robot's affective state. High values of e cause the robot's affective state to align closely with the user's state, whereas low values preserve the robot's affective state. An important point already illustrated by the state machine, in Figure 18, in Section 3.4.2.1, representing ROSE's dynamic affective state-change model is that, the moment PULSE sends ROSE new information regarding a change in the user's affective state, the transformation of that state occurs instantaneously, such that $\mathbf{s}_r = \mathbf{s}'_r$.

4.1.2.2 Emotional Decay Instantiation

Emotional decay is the ROSE extension point that aims to provide a mechanism for the robot's affective state to progressively return towards the base mood $\mathbf{b}$ defined in the Robot Affective Profile settings and which seeks to represent a persistent affective tendency of the robot's affective state. For this instantiation, a linear interpolation will also be used, however, it is calculated iteratively during the feedback process by the affective dynamic model. To this instantiation the state is updated as

$$\mathbf{s}_r(t + \Delta t) = (1 - d) \cdot \mathbf{s}_r(t) + d \cdot \mathbf{b}, \quad (4.6)$$

where $d \in [0,1]$ is the emotional decay factor. Emotional decay is continuously executed by ROSE's dynamic affective model until the robot's affective state converges to the base mood. Its execution begins after a time interval ℓ , the value of the decay delay parameter, also defined in the Robot Affective Profile, allowing the robot to remain in

the user's affective state during that period.

Both the dynamic affective model and the variables defined in the Robot Affective Profile aim to establish robotic profiles capable of simulating dynamic affective behaviors, which can be used for exploratory purposes in research on affective human-robot interaction. However, they do not in any way represent a realistic model of affective profiles or the transitions between emotions in human beings.

Those equations represent one possible implementation of the ROSE extension points.

4.1.3 AEB Instantiation

The approach used for the mapping process for this instantiation was designed to support the demonstration scenario presented in the following section. Euclidean distance calculation was used to determine the emotion during the mapping process for the vector $\mathbf{s}_r$. Due to these characteristics, the presented method was named Reference Point-Based Empathic Behavior Mapping. The following section describes in detail the options chosen for this instantiation.

4.1.3.1 Empathic Behavior Mapping Instantiation

The method employed maps the vector $\mathbf{s}_r$ to discrete categories, specifically anger, happiness, sadness, and neutrality, as these emotions proved easiest to classify and typically yielded higher accuracy during the classification process, particularly in FER models. Since the proposed output is discrete and categorical, intensity points ranging from the axis center to the anchor points were defined to assign some levels to these emotions. Each emotion comprises levels L1, L2, L3, and L4, with the final level corresponding to the anchor point itself. This approach uses the same anchor points for the emotions of the VAD space employed in Section 4.1.1.2.

It is important to note that the abstractions regarding the levels of emotion (L1–L4), the positioning of their representative elements, as proposed in this instantiation, consists of values intended only to demonstrate the framework's extensibility and its application within the scenario described in the following section. To make it easier to visualize, Figure 27 shows the graph with the 2D projection of the anchor points for the selected emotions and the defined intensity levels. The points (L4) shown in the figure represent the anchor points for these emotions, used as a reference in the CAT-to-VAD process described in Section 4.1.1.2. Although the figure does not show a three-dimensional graph, only a 2D

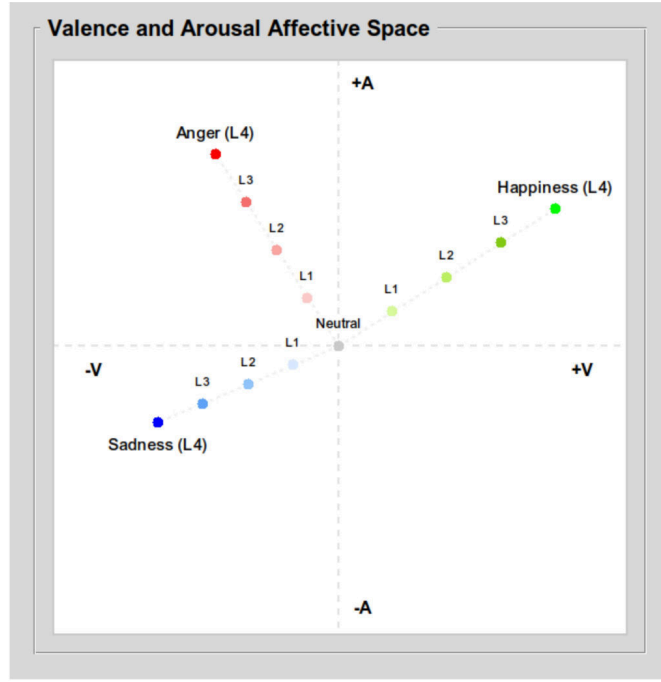


Figure 27: 2D projection of anchor points in the VA plane for the emotions of anger, happiness, sadness, and neutral, and the intensity level points (L1–L4) defined for this instantiation.

projection of the robot vector, all calculations involving the robot and user vectors are performed using the three dimensions of the VAD space.

The mapping of the vector $\mathbf{s}_r$ is performed by calculating the Euclidean distance $d_i = \sqrt{(V_r - V_i)^2 + (A_r - A_i)^2 + (D_r - D_i)^2}$ between the vector's VAD components and each points i belonging to the set of anchor points and their intensity level points. The point closest to the vector characterizes the robot's emotion and its corresponding level.

With the returned value in an "emotion_level" format, the process moves to the next stage: generating empathic behavior. Since this instantiation is intended for the FRED robotics platform, the output data structure p will be formatted taking into account the platform's capabilities and limitations. Based on this, this instantiation directly maps the "emotion_level" format to four characteristics: affective_state, facial_expression, leds, and pose. An example of the generated structure looks like this:

$$p = \left\{ \begin{array}{l} \text{affective_state : happiness_L3,} \\ \text{facial_expression : L3,} \\ \text{pose : L3,} \\ \text{leds : L3} \end{array} \right\} \quad (4.7)$$

At the end of the process, the structure is sent to the Execution Layer, published as an MQTT message. The message is received by the Shared State Sync component and passed to the robot's memory. The *setter* method responsible for storing the AEB behavior information parses the structure, saving each element with its key and value, prepending "aeb_" to each key name, and stores them as system variables. These variables can be accessed by SENSEI and have their values compared during execution, allowing these multimodal parameters to be triggered by the corresponding EvaML command. This makes it clear that the AEB is not responsible for executing the behavior, leaving that responsibility to the Execution Layer.

All the choices made, such as the types of representation, the dimensionality reduction used, and the output structure, are specific to this instantiation example and do not limit this extension point to any of the approaches employed here.

4.1.4 Affective Interaction Scenarios

This section presents an affective interaction scenario that integrates the instantiations of the affective components, described in previous sections, with the Execution Layer and the affective robotic platform. During the interaction, the user's affective state is estimated via Text Emotion Recognition (TER) and Facial Expression Recognition (FER), these data are processed by PULSE, transformed by ROSE according to the active Robot Affective Profile, and mapped by AEB into behavioral parameters. These parameters are then made available to SENSEI, which coordinates their use during the execution of the EvaML interaction script on the robot. The same interaction is performed using two distinct Robot Affective Profiles to illustrate how different affective configurations can result in different robot behaviors.

4.1.4.1 Interaction Setup

The interaction scenario was executed on an Acer Swift 3 notebook with 8 GB of RAM and a 512 GB SSD, without a dedicated GPU. Although it is not the focus of this thesis, some basic information about the multimodal modules used for text emotion recognition (TER) and facial emotion recognition (FER) will be described. The laptop builtin microphone and webcam were used to capture voice audio and for FER, respectively, thus, the only external hardware used during the video interaction was the robot itself.

The FER system was implemented using the DeepFace framework ([SERENGIL;](#)

OZPINAR, 2021), with OpenCV (BRADSKI, 2000) employed for image capture and processing. To avoid overloading the computer, a capture rate of 5 frames per second was set for each affective perception window, and each classification was stored in a list of observations. For each of the seven categories (neutral, fear, disgust, happiness, sadness, angry, surprise), the module calculated the median of the collected observations. Subsequently, these resulting values were normalized. Finally, this CAT vector was sent to PULSE.

TER was performed using the *michellejeili/emotion_text_classifier* (HARTMANN, 2022) model through the Hugging Face Transformers library. It follows the same strategy with the calculation of medians for the probabilities of each of the seven categories, which are normalized and sent. In the case of TER, instead of classified frames, the list of observations must store the classifications of each sentence within the current perception window. In the interaction shown in the video, only one sentence per perception window was processed.

4.1.4.2 Interaction Flow

This section aims to achieve the following goals: to show the functioning of the framework's affective components by executing the previously presented concrete implementations; to provide evidence that the thesis proposal enables the construction and exploration of interaction scenarios in the field of affective human-robot interaction, serving as a support tool for researchers; and to demonstrate that modifying the Robot's Affective Profile parameters allows it to react differently when receiving the same multimodal stimuli from the user.

In both interactions, the user's goal is essentially to produce the same multimodal stimuli, making it possible to observe how the robot reacts to them based on each of the two affective profiles employed. To facilitate understanding, two figures are provided that offer a detailed representation of the interaction flows shown in two demonstration videos. Each interaction was divided into seven numbered segments, displaying the dialogue between the user and the robot, their affective states and corresponding VAD vector values, as well as the start and end of the affective perception windows and the multimodal inputs selected for each of them.

Scenario No. 1 – Empathetic Affective Robot Profile. The first scenario, presented in Figure 28, uses a robot with a profile designated as "empathetic." In studies involving

robots capable of recognizing user emotions during interaction, it is common for the robots to be programmed to "mimic" the user's emotion. The term "empathic" is borrowed here from psychology, without claiming to strictly represent its precise definition, the term is used simply to describe a robot that responds to user interaction with behaviors reflecting ("imitating") the user's own emotions.

The graphical representation of the first interaction scenario can be seen in Figure 28. The interaction is described in the figure from top to bottom. It begins by showing the name of the affective profile used and its parameters, the same ones described in the ROSE definition. The first parameter, the empathy factor, has a value of 1.0 (100%). Based on the equation used for the instantiation, this means that in the empathy-driven transformation performed by ROSE, 100% of the vector $\mathbf{s}_u$ (representing the user's affective state) is retained, without taking the robot's vector $\mathbf{s}_r$ into account. It is this factor that makes the robot "empathetic." The base mood vector, which defines a persistent affective tendency for the robot, places it at $\mathbf{s}_r = \begin{bmatrix} 0.2 & 0.1 & 0.0 \end{bmatrix}^T$, which, based on the 4 emotion levels defined in the instantiation in the framework, categorically corresponds to the "Happiness L1" level. The influence of the other parameters, the emotional decay factor and decay delay, can be better observed in the video linked at the end of this section, as these parameters directly affect the response of the robot's affective dynamics model regarding its return to the base mood state.

To facilitate the description of the interaction flow, the image was numbered from 1 to 7. The text and heart icons were colored to serve as a legend for each emotion, with the absence of a heart icon representing neutrality. Red was used for anger, green for happiness, and blue for sadness. Furthermore, the emotions of the robot and the user are displayed as values on a categorical scale (based on the 4 emotion levels defined in the instantiation in the component), and in the VAD model, within a column vector $\mathbf{s} = \begin{bmatrix} s_v & s_a & s_d \end{bmatrix}^T$.

In (1), the robot is in a base mood ($\mathbf{s}_r = \begin{bmatrix} 0.2 & 0.1 & 0.0 \end{bmatrix}^T$) and the user is neutral ($\mathbf{s}_u = \begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$). The robot then greets the user politely.

In (2), executing the `<perception>` command triggers a "START" action, initiating an affective perception window that utilizes TER and FER sources, as shown in the image. At this point, the FER module captures and analyzes video frames while the user speaks a sad sentence, expressing the corresponding emotion through facial expressions. The spoken audio is captured by the laptop's microphone and converted into text, which is then passed to the TER module. The `<perception>` command sends an "END" action, instructing

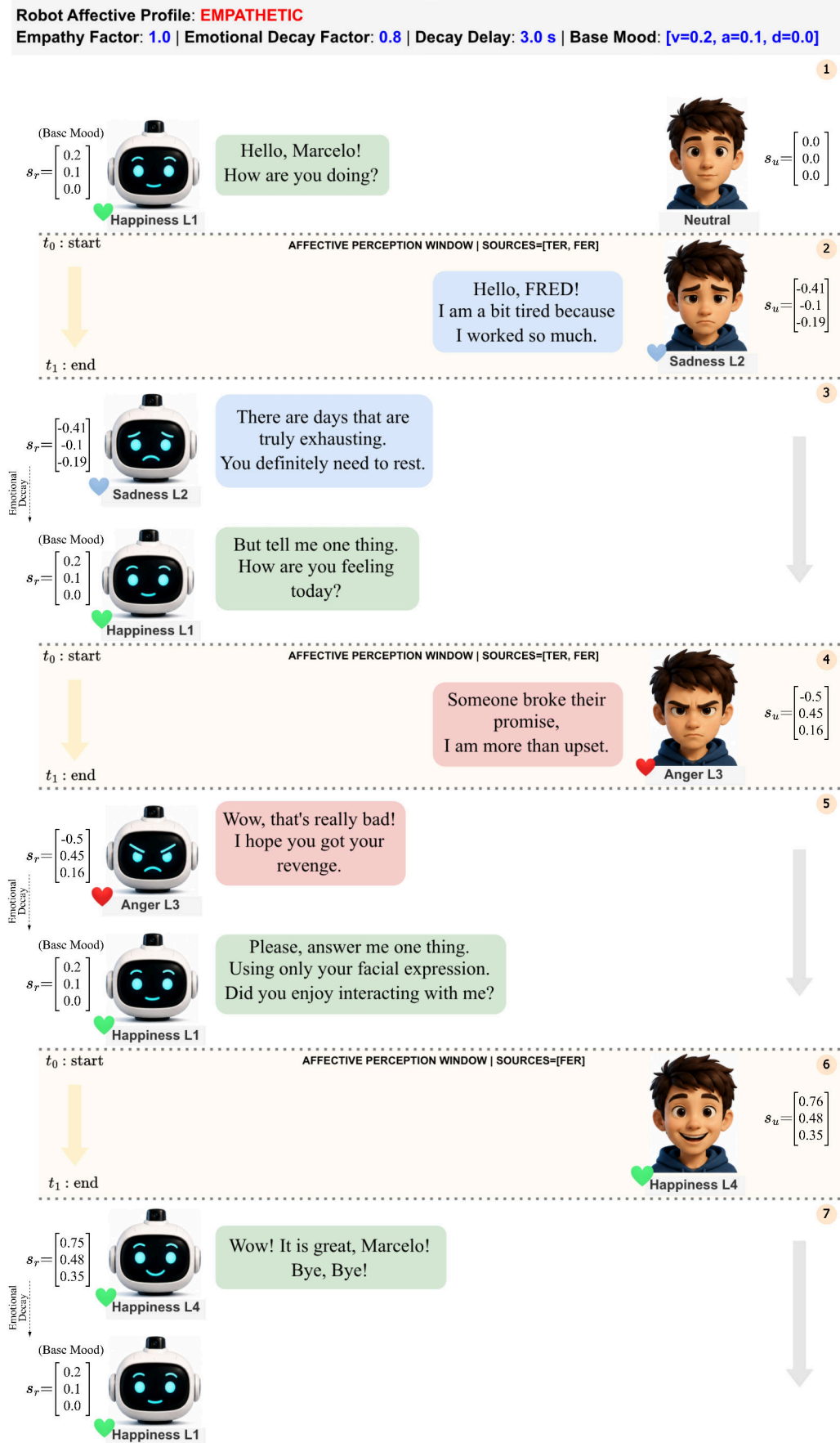


Figure 28: Scenario 1 - Interacting with an "empathetic" affective robot.

both the TER and FER modules to process their data and send their classification results to PULSE. PULSE calculates the confidence levels of the multimodal module outputs, converts CAT values to VAD, and finally performs multimodal fusion using a weighted sum based on the previously obtained confidence values. As shown in the figure, the user was classified, on a categorical scale, as "Sadness L2" and, in the VAD emotional space, by the vector $\mathbf{s}_u = \begin{bmatrix} -0.41 & -0.1 & -0.19 \end{bmatrix}^T$. This vector is then sent to ROSE.

Upon receiving the user's affective state vector (3), ROSE applies an empathic transformation based on the empathy factor value defined in its profile, thereby adopting, as expected, the same affective state as the user: $\mathbf{s}_r = \mathbf{s}_u$. This information is passed to the AEB component, which performs a mapping, categorizes the emotion according to specific levels (L1–L4), and generates a key-value structure containing the multimodal elements appropriate for the robot featured in the video. Finally, this structure is sent to the Execution Layer. Using the values provided by the AEB, the EvaML script controls the robot's multimodal elements, generating the empathic behavior. Based on these values, the robot expresses sadness through its gaze and the blue LEDs on its chest, while saying the text specified in the script for that situation. After uttering the sad words, the robot's emotional state returns to its base mood, and it asks another question.

In (4), another affective perception window starts, also using TER and FER as inputs. This time, the user expresses anger in both their spoken text and facial expression. The steps of capture, perception window termination, sending to PULSE, and user affective state calculation are repeated. In this case, $\mathbf{s}_u = \begin{bmatrix} -0.5 & 0.45 & 0.16 \end{bmatrix}^T$, which corresponds to "Anger L3". The same sending process for ROSE occurs again.

In (5), ROSE receives the vector and processes it, causing the robot's affective state to match the user's. The process of sending the vector to the AEB, mapping it, and forwarding it to SENSEI repeats, enabling the EvaML script to verify the parameters sent to the AEB and then execute an expression of anger while speaking the corresponding pre-defined phrase. Once again, the robot's affective state is updated to the base mood. The robot then asks the user a question but requests a response only through facial expression. This was done to demonstrate that it is indeed possible to select inputs using the [source](#) attribute of the `<perception>` command.

Another perception window starts at (6), but now only FER is used as input. In previous cases, the perception window duration was implicitly defined by the blocking caused by the voice capture command, `<listen>`. In this instance, the FER capture duration (3 s) was explicitly defined using the `<wait>` command. After the 3-second interval,

the entire process repeats until PULSE indicates that $\mathbf{s}_u = [0.76 \ 0.48 \ 0.35]^T$, which categorically corresponds to "Happiness L4", a high level of happiness. This information is sent to ROSE.

Upon receiving the user's affective state (7), ROSE processes it, resulting in $\mathbf{s}_r = \mathbf{s}_u$. The entire post-ROSE process is repeated, the robot becomes overjoyed, matching the user's state of $\mathbf{s}_r = [0.75 \ 0.48 \ 0.35]^T$, bids a cheerful farewell, and returns to its base mood, concluding the interaction. Upon closer inspection, there was a slight difference between the valence component values in $\mathbf{s}_r = [\mathbf{0.75} \ 0.48 \ 0.35]^T$ and $\mathbf{s}_u = [\mathbf{0.76} \ 0.48 \ 0.35]^T$, during stage (6). It was not a typo, it was caused by rounding issues during the empathy-driven transformation.

Scenario No. 2 – Optimistic Affective Robot Profile. Since a significant portion of the processes involved in the interactions was already detailed for Scenario No. 1, only the points differing from the first interaction will be described in detail here. In this scenario, presented in the Figure 29, the same robot was used, interacting with the same user, speaking the same phrases, and expressing the same emotions through its face. What changes, then, is the robot's affective profile, labeled "optimistic" in this case. The concept is that, unlike in the first scenario, the robot would not place as much weight on the user's affective state, remaining closer to its own persistent affective tendency, or base mood. Its profile was termed "optimistic" because its base mood coordinates place it categorically in "Happiness L2" and because it utilizes only 40% (empathy factor = 0.4) of the user's affective state during the empathic transformation.

In (1), the same process as in (1) of the previous scenario is repeated, differing only in the base mood value, $\mathbf{s}_r = [0.4 \ 0.3 \ 0.2]^T$.

In (2), the same process occurs as in (2) of the first scenario: PULSE arrives at a result categorically identical to the user's ("Sadness L2"), but with a slight difference in the VAD vector components. Next, PULSE sends the result to ROSE.

The difference in the robot's affective profile appears now, in (3): instead of becoming sad, as happened with the same item in the previous scenario, after ROSE processing, the robot shifts to "Happiness L1" with $\mathbf{s}_r = [0.08 \ 0.2 \ 0.08]^T$. It remains in the first quadrant, exhibiting low valence but arousal equals 0.2. After the data is sent to the AEB and subsequently to SENSEI, the robot responds to the user's sadness with a smile, green LEDs and a motivational phrase. The robot then returns to its base mood and asks the

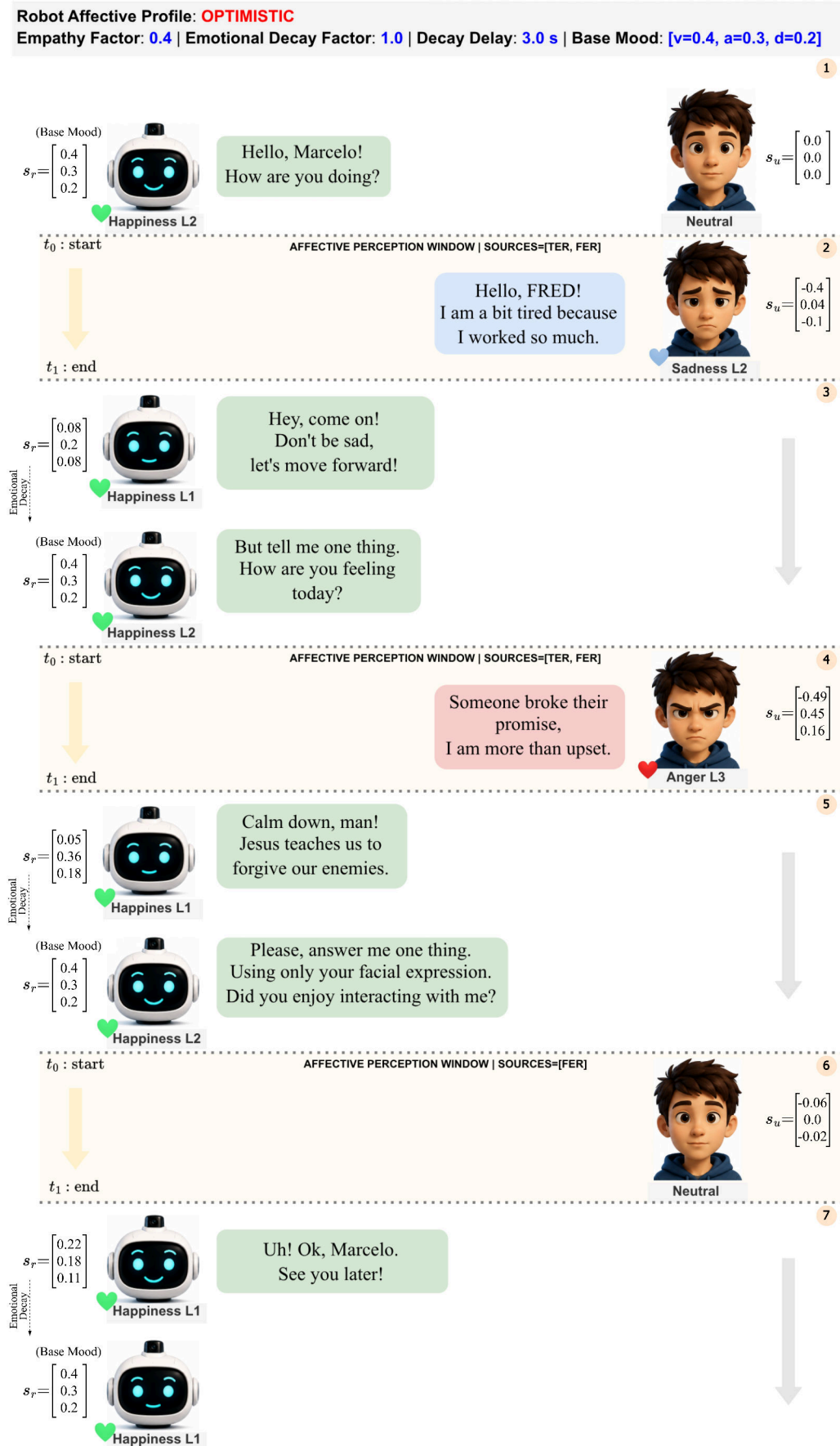


Figure 29: Scenario 2 - Interacting with an "optimistic" affective robot.

next question.

Item (4) is repeated with the same item from the previous scenario, with PULSE calculating the user's affective state result as $\mathbf{s}_u = [-0.49 \ 0.45 \ 0.16]^T$, which is categorically equivalent to "Anger L3". So, PULSE sends the value to ROSE.

In (5), once again, consistent with the objective dictated by the affective profile configuration following ROSE and AEB processing and the transmission to SENSEI, the robot remained categorically closer to "Happiness L1" ($\mathbf{s}_r = [0.05 \ 0.36 \ 0.18]^T$). Consequently, based on this state, it was able to maintain a slight smile, display green LED lighting, and speak the predefined phrase, which encouraged the user to remain calm and fostered forgiveness, rather than inciting revenge as in the previous scenario. Afterward, the robot returned to its base mood and asked the question that required only a facial expression as a response.

In (6), the perception window is started again, taking only FER as input. PULSE yields the result $\mathbf{s}_u = [-0.06 \ 0.0 \ -0.02]^T$, which corresponds categorically to "Neutral." This information is sent to ROSE.

In (7), after the ROSE processing, the robot moves to $\mathbf{s}_r = [0.22 \ 0.18 \ 0.11]^T$, remaining categorically close to "Happiness L1". ROSE sends the vector to the AEB, which processes it and forwards it to SENSEI. Based on the value received from the AEB, the script then expresses the emotion level through the defined multimodal elements and speaks the predefined phrase, bidding farewell, returning to the base mood, and concluding the interaction.

Having detailed the two scenarios, it is now appropriate to explain how the EvaML script processes the multimodal expression parameters defined in the key-value structure sent by the AEB. As previously stated in the section introducing the AEB, the component itself is not responsible for executing the expressions and behaviors on the robot. Its function is only to perform the mapping process, structure the data, and send it to the Execution Layer, which is the component actually responsible for carrying out actions on the robot.

Taking the EvaML script from scenarios 1 and 2 as an example, a macro named "AEB_EXECUTE" was created. This macro checks the system variables, which are generated upon receipt of the key-value structure, and links them to the generated parameter values such as {pose: L3} for the EvaML command directed at the robotic platform. The macro is invoked whenever the script author wants the robot to express itself based on the

parameters sent by the AEB component. This approach allows for greater control over the use of parameters automatically generated by the AEB, making it possible, for example, to prevent the robot from displaying state changes while executing the affective dynamic model that automatically returns it to its "base mood." Such control over behavior is crucial in situations requiring precise management of the robot's actions and expressions.

Listing 4.1 presents a snippet of the macro defined to execute scenario interactions. For the sake of simplicity, and only for the purpose of recording the demonstration video, only the system variable `aeb_affective_state` was used in the verification process and in establishing the relationship between parameters received from the AEB and the robotic platform's functionalities. In addition to the snippet showing the macro definition, the code also displays the selection of the Robot Affective Profile within the `<settings>` section, as well as a snippet showing control of the Affective Perception Window through the `<perception>` command and its `action` and `source` attributes.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <evaml name="scenario_1"/>
3   <settings>
4     <voice type="en_US-bryce-medium" />
5     <lightEffects mode="ON" />
6     <audioEffects mode="ON" />
7     <robotAffectiveProfile profile="EMPATHETIC" />
8   </settings>
9   <script>
10    <talk>Hello Marcelo!</talk>
11    <talk>How are you doing?</talk>
12    <!-- Opens the APW and selects the [TER, FER] sources -->
13    <perception action="START" sources="TER, FER"/>
14    <!-- Captures the user's response and returns a string -->
15    <listen var="USER_RESPONSE"/>
16    <!-- Sends the user response (string) to the TER input module -->
17    <mqtt pubTopic="TER_TEXT"> #USER_RESPONSE </mqtt>
18    <!-- Ends the perception window -->
19    <perception action="END" />
20    <!-- Executes the robot behavior generated by the AEB. -->
21    <useMacro macro="AEB_EXECUTE" />
22    .
23    .
24  </script>
25  <macros>
26    <macro id="AEB_EXECUTE">
27      <switch var="aeb_affective_state">
28        <!-- #### Happiness affective state #### -->
29        <case op="exact" value="happiness_l1">
30          <light state="ON" color="GREEN"/>
31          <evaEmotion emotion="PLEASED" />
32          <led animation="HAPPY1" />
33          <pose type="NEUTRAL" />
34        </case>
35        .
36        .
37        <case op="exact" value="happiness_l4">
38          <light state="ON" color="GREEN"/>
39          <evaEmotion emotion="HAPPY" />
40          <led animation="HAPPY4" />
41          <pose type="LEFT_FOOT1" />
42        </case>
43      </switch>
44    </macro>
45  </macros>
46 </evaml>

```

Listing 4.1: A snippet of the EvaML code from the AEB_EXECUTE macro, responsible for executing the behaviors generated by the AEB on the robotic platform.

Another aspect concerns the phrases spoken by the robot during interactions. Although that were defined to better convey the robot’s emotion within the specific context of use, nothing would prevent these phrases from being generated by an LLM, with both the phrases and a system prompt being sent within the AEB structure. Both EvaML and the framework components support the requirements needed to implement this.

The video presenting the two scenarios of affective interaction between the user and the robotic platform can be viewed at the following [link](#). During the interaction, the video displays the laptop webcam feed capturing the user’s face, the image of the robotic platform, two horizontal bar charts showing real-time probability values for each emotion, and a small auxiliary application designed exclusively to visualize, on a 2D (VA) plane, the positions of the robot’s and user’s affective states as they interact.

This section provided concrete implementations for the extension points defined by PULSE, ROSE, and AEB. These implementations represent specific choices made to demonstrate the functionality and integration of the proposed components. Furthermore, two affective interaction scenarios were proposed, which showed evidence that affective components can, in fact, abstract the necessary requirements for their application in research on Affective Human-Robot Interaction, serving as a support tool for researchers. A link was also made available to a video that demonstrates these scenarios during an interaction between a user and a robotic platform. The execution of these scenarios demonstrates the integration of the three layers of the framework: the Affective Layer (PULSE, ROSE and AEB), the Execution Layer (SENSEI) and the RAL (FRED robot), whose instantiation will be presented in Chapter 6, Section 6.2. These scenarios offer a functional demonstration of the proposed approach to address Research Question 2, showing how affective information obtained from different sensing sources can be processed and integrated by PULSE to estimate the user’s affective state during an interaction. Furthermore, the video shows how the same robot, configured with different affective profiles, can behave differently in response to the same affective state of the user.

The following section presents the instantiation of the proposed framework on the EVA robotic platform and describes the resulting extensions to the robot’s non-verbal communication and multimodal interaction capabilities.

4.2 EVA Robot 2.0

This section demonstrates the instantiation of classes used to create the control software for the EVA robot 2.0, including all its capabilities, and the simulator EvaSIM 2.0, which is capable of executing EvaML scripts directly on the physical robot.

4.2.1 Instantiation of EVA Robot 2.0

As previously mentioned in the Chapter 3, the control modules defined using the framework use the publish-subscribe paradigm of asynchronous message exchange. Each robot control module, which implements each of its functionalities, is controlled through messages that are defined by the system developer. These messages are also used as a synchronization mechanism between modules, in addition to allowing the sharing of information related to their services. The values coming from the readings of sensors such as a microphone (in the specific case of a module that transforms audio into text) or from a module capable of classifying the emotion contained in a text can be passed from one module to another through the exchange of messages published in specific topics intended for this purpose.

The EVA 2.0 control software is composed of 10 modules. Nine of them run inside a *Raspberry PI 4* board, which is embedded inside the 3D printed body of the robot. Those tenth module runs on a separate machine. The nine modules are responsible for controlling each of the functionalities. All modules were implemented based on the framework in this work, more specifically, through the use of the elements available in the RAL. Figure 30 provides an overview of these modules and the communication scheme of each of them with the MQTT broker.

Based on the behavior of each module of the EVA 2.0 system, it is possible to divide them into three distinct groups: I) modules that perform tasks asynchronously, without using any synchronization mechanism with other modules, that is, they receive an activation command and perform their tasks without signaling their completion; II) modules that perform synchronous tasks and return some type of response (in text format) signaling their completion through an unlock message; III) modules that perform tasks synchronously, as well as modules in group II, but without returning any type of response in text format. Since the instantiation processes of the modules belonging to each group are similar to each other, these processes will be described and discussed based on each of the three groups. Table 4 presents each group and its modules.

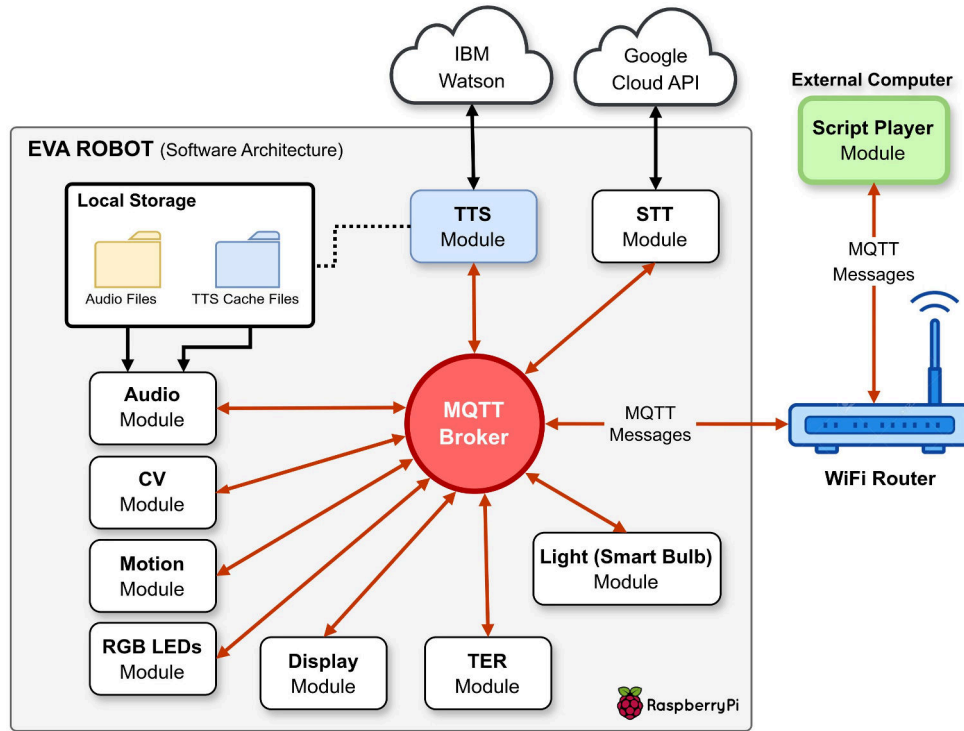


Figure 30: Software architecture of the EVA 2.0 implemented from the proposed framework.

Table 4: Topics and messages for the implementation of the EVA 2.0 software.

Group	Module	Tópico (Subscribed)	Message
I	Display	evaEmotion	"HAPPY", "FEAR", "SAD" etc.
	Light	light	"BLACK OFF", "BLUE ON" etc.
	RGB LEDs	leds	"HAPPY", "ANGRY" etc.
	TTS	talk	"Voice Text"
	Motion	motion/head	"UP", "DOWN", "LEFT" etc.
		motion/arm/left	"POS0", "POS1", SHAKE1 etc
		motion/arm/right	"POS0", "POS1", SHAKE1 etc
II	STT	listen	Empty text
	TER	textEmotion	"Text"
	Computer Vision	userEmotion	Empty text
		userID	Empty text
		qrRead	Empty text
III	Audio	sound	"file_name BLOCKING"
		speech	"file_name"
	ScriptPlayer	"See Table 5"	"See Table 5"

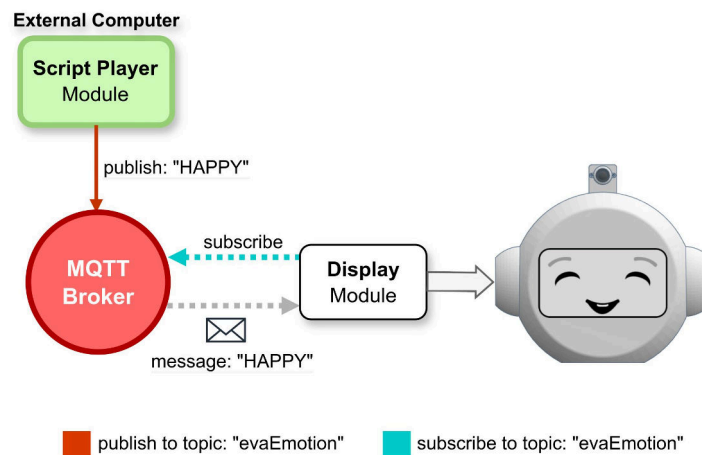


Figure 31: Controlling the robot's gaze expressions by publishing the message "HAPPY" in the topic *evaEmotion*.

4.2.1.1 Group I Modules

The first group is formed by the Motion modules, responsible for controlling the movement of the robot's head and arms, by the RGB LEDs module, which executes animations with the RGB LEDs on the EVA's thorax, by the Light module, which controls the Smart Lamp, and by the Display module, which shows the robot's gaze expressions through a 5.5" display. All modules in this group begin executing their tasks upon receiving an activation command from another module. Activation commands are specific messages published in the broker in topics determined by the system developer. After execution, a module in group I does not return any information as a response nor does it use any mechanism that could signal the end of its task to the module that controls it. A robot control module, like the Script Player module, which will be covered later. You can then send a command to the Display module, causing the EVA to show a happy expression. The control module achieves its goal by publishing the message "HAPPY" in the *evaEmotion* topic, previously subscribed to by the Display module. Figure 31 shows the entire process described along with the elements involved.

Figure 32 shows the instantiation of the Display module. Basically, to implement a module using the proposed framework, it is necessary that the module be developed following the three steps:

1. Extend the abstract class **RobotModule**. By doing so, the module becomes able to define a base topic that will be used as the root topic of the robot system. In addition, it becomes able to instantiate a connection object with the MQTT broker.

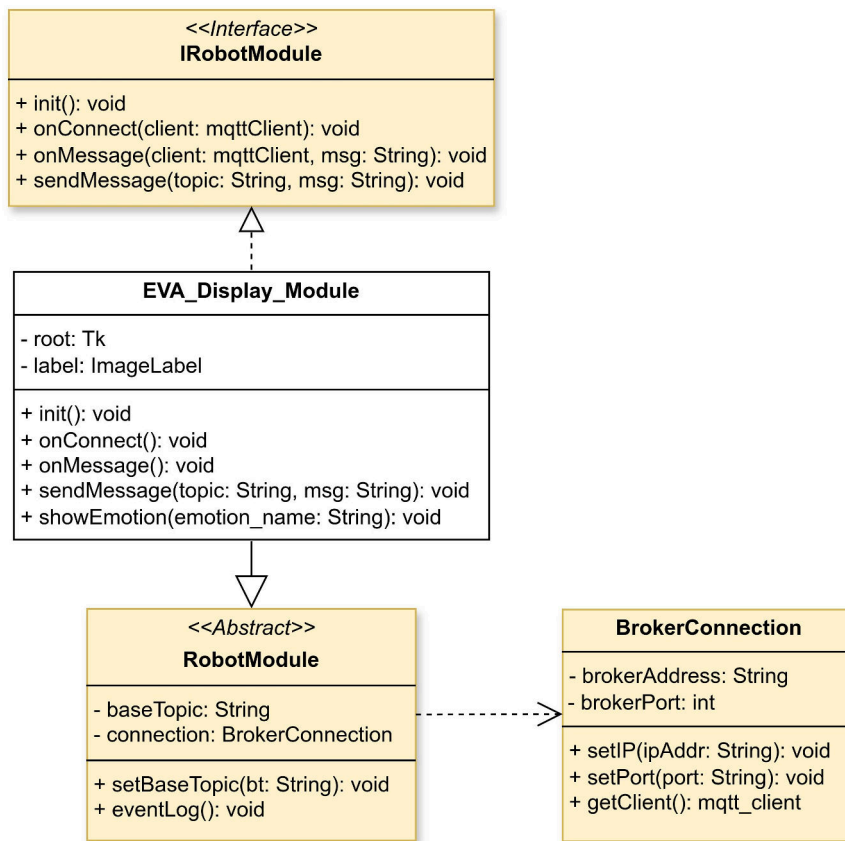


Figure 32: Instantiation of the Display module.

2. Instantiate the connection object with the broker by obtaining an object of type *mqtt_client*. This allows the module to be able to "subscribe" and/or "publish" to the topics of its interest.
3. Implement the **IRobotModule** interface. The interface contract determines that every module that intends to implement it must implement the methods: *onConnect()*, which is where the module must subscribe to the topics of its interest; the *onMessage()* method, which is where the module receives and handles the messages published in the topics it subscribes to; the *sendMessage()* method, which must implement the sending (publishing) of messages, if the module returns a value and, finally, the *init()* method, which must be implemented in order to instantiate the connection object with the broker, obtaining the MQTT client and attaching to it the aforementioned callback functions: *onConnect()* and *onMessage()*.

The other modules in group I, since they present the same behavior pattern, follow the same instantiation process as the Display module.

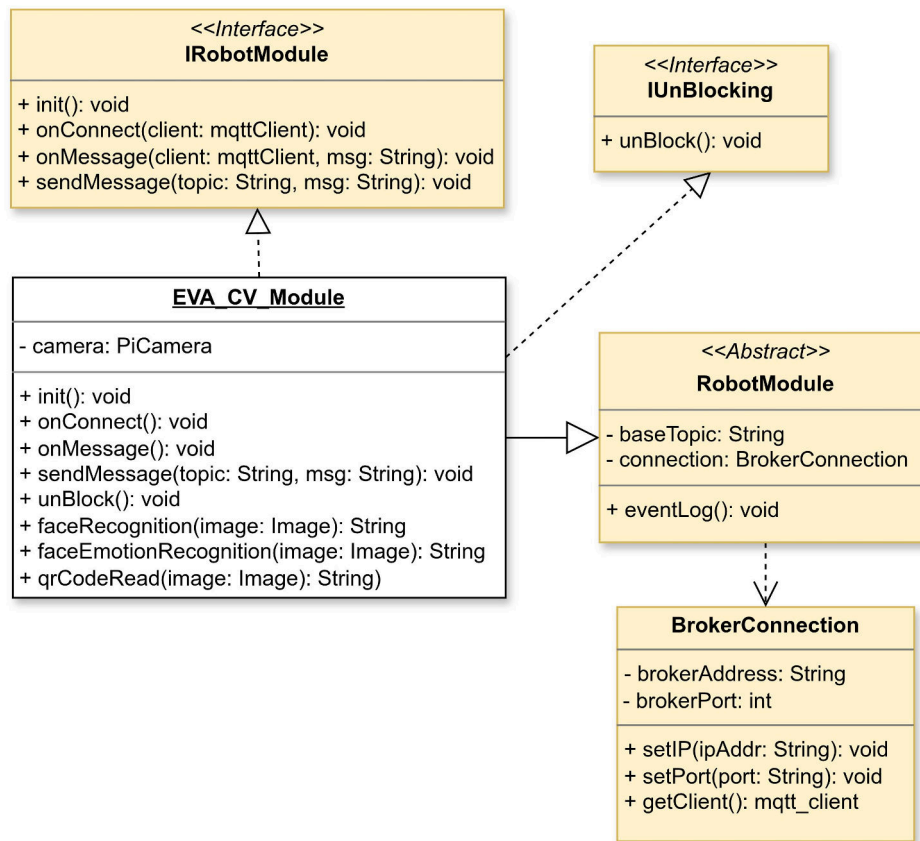


Figure 33: Instantiation of the CV (Computer Vision) module.

4.2.1.2 Group II Modules

The modules that are part of group II are: the STT (Speech-To-Text) module, which returns a text from the user's speech audio, the CV (Computer Vision) module, which executes three functionalities based on images captured by the PiCamera, returning its responses in text format, and the TER (Text-Emotion-Recognition) module, which classifies a text returning the emotion contained in it. The response returned by the TER module is also in text format. The instantiation of the CV module can be seen in Figure 33.

The EVA computer vision module performs three tasks based on images captured by the PiCamera, which is attached to its head. The camera's driver does not allow it to be shared by more than one process simultaneously, meaning that the user's face identification, facial expression recognition and QR Code reading functions, which use the camera, must be implemented in a single module, that is, in the module that connects to the camera.

A module belonging to group II, like the modules in the first group, must have the

Table 5: Special topics and messages used in the EVA software.

Purpose	Topic	Description
Event log	log	EVA robot modules use this topic to provide information about their states and services.
Passing variable values	var	This topic is used to pass variable values between modules. It is used by the CV, STT, TER and Script Player modules.
Inter-module synchronization	state	Through this topic, modules can work synchronously, exchanging lock and unlock messages.

basic characteristics necessary to implement a function in the robot using the proposed framework. To do this, the CV module follows the same implementation steps as the modules in group I. The modules in group II can perform tasks in a synchronized manner with the other modules, signaling the end of their activity through an unlocking mechanism (a message). To achieve this, the proposed framework provides an interface called **IUnBlocking** that must be implemented by the module. This method must be responsible for sending (publishing) the message that signals the end of the execution of the module being implemented. All EVA modules that implement this method send their synchronization messages to a special topic, called *state*. In the EVA software, every module that controls or is controlled by another module uses the publication of messages on this topic as a synchronization mechanism. In addition to using the synchronization mechanism, with the implementation of the **IUnBlocking** interface, a group II module always returns some type of information that comes from its service and this information is always in text format (a string). In the EVA system, modules that need to pass their information to other modules use the `sendMessage()` method. This method is used to send/publish the information to be returned in the form of a message through the special topic *var*. Therefore, any module interested in receiving the value of a variable must subscribe to this topic. Table 5 shows, in a more detailed manner, the list of special topics used in the EVA system.

4.2.1.3 Group III Modules

The EVA modules that belong to group III are capable of synchronizing their execution with other modules. They implement the **IUnBlocking** interface and are capable of signaling the completion of their tasks by sending/publishing messages in the special topic, *state*. However, they do not return any type of textual information. As can be seen in Table 4, the modules in this group are: the TTS module, the Audio module and the

Script Player module.

The TTS module is responsible for transforming a text (a string) into an audio file. This file contains the audio of the text spoken by one of the voices provided by the IBM Watson cloud service. The module is activated upon receiving a message published in the *talk* topic. As can be seen in Table 4, the message must contain the specification of the voice to be used in the TTS process, concatenated with the text that must be transformed. As can be seen in Figure 30, the audio files generated by IBM Watson are stored in a local folder called *TTS Cache Files*. After the audio file generation process, the *TTS* module sends an activation command to the Audio module, which must play the audio file generated by Watson and stored in *cache*. This command is a message published in the *speech* topic, which is subscribed to by the Audio module and contains the name of the audio file with the speech.

The Audio module is responsible for playing all types of audio used by the robot, be it a music file, a sound effect or an audio file generated by the TTS module. As can be seen in Table 4, the module subscribes to two topics, the *sound* topic and the *speech* topic, and is activated by receiving messages published in these topics. The first of these, *sound*, receives in the message text the name of the audio file to be played, concatenated with a *boolean* information ("TRUE" or "FALSE") that indicates whether the module should run synchronously or asynchronously with the control module. If the value passed is true, the module will send an unlock signal when the audio playback is finished. The second topic subscribed to by the Audio module is *speech*. The messages received in this topic activate the module and contain only the name of the speech audio file, which was previously stored in the local folder *TTS Cache Files*. All speech audio files are played synchronously, signaling their completion through the special topic, *state*.

The Script Player module (SP) has all the features of the other two modules in group III, but it has some unique features. The SP module, which implements EvaSIM 2.0 simulator, is a control module, capable of reading interaction scripts written in the EvaML language. The SP module can read the content of an XML file containing the EvaML script and execute commands that represent the robot's functionalities, such as: `<evaEmotion>`, `<light>`, `<motion>`, `<talk>` etc. When reading each command from the script with its respective attributes, the SP module sends activation signals to the respective modules, causing the robot to respond to the commands being sent. In order to execute commands in a synchronized manner, the SP module implements another interface provided by the framework, the **IBLocking** interface. By implementing the

block() method, the SP module can block its execution flow, activate a module that works synchronously like the STT module and, after the execution of the module, which signals its completion by sending a message to the special *state* topic, the SP module is unblocked, continuing its flow by executing the next command.

To facilitate the implementation of modules that need to implement some type of memory for the robot, the framework provides an abstract class that offers a simple memory model with some concrete methods implemented. This module can have its methods overloaded in order to provide the manipulation of other types of data, in addition to Strings. Figure 34 shows the **IBlocking** interface, the abstract class **RobotMemory** and the instantiation process of the three modules of group III.

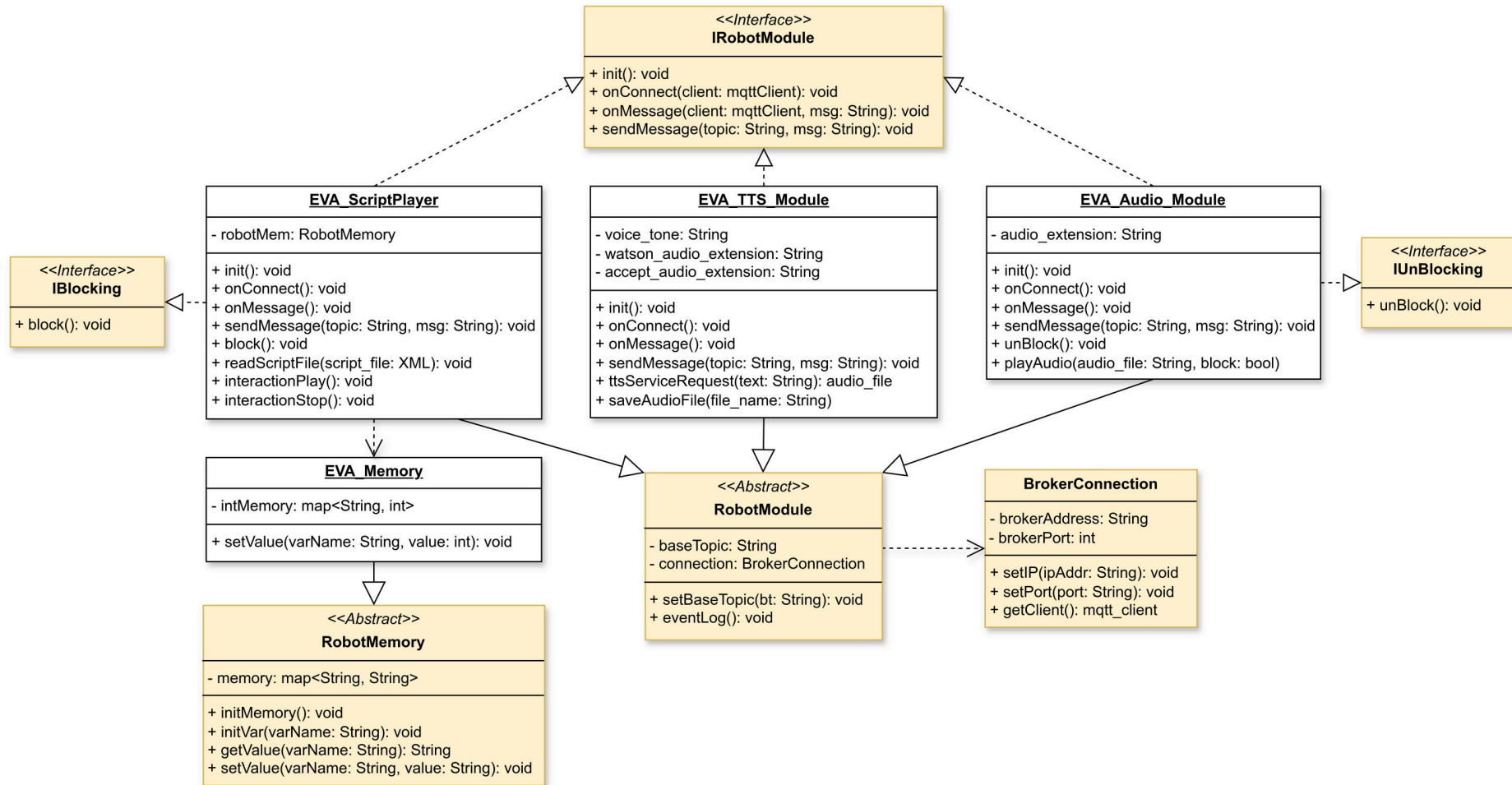


Figure 34: Instantiation of the TTS, Audio and Script Player modules.

Figure 35 presents a sequence diagram that demonstrates how a TTS command is executed by the modules SP (from its sending), TTS (the text-to-speech transformation process) and Audio (playing the speech audio file). By following the sequence presented in the diagram, it is possible to understand more clearly the synchronization process between modules.

When executing an EvaML *script* and encountering a command of the `<talk>` type, the SP module publishes a message to the *talk* topic. As specified in Table 4, the message must contain the voice timbre that will be used by the IBM Watson service and the text to be transformed to speech. The SP module then calls its *block()* method, blocking its execution flow. The module will remain blocked until an unblocking message is published to the special topic, *state*. Upon receiving the message, the TTS module is activated, immediately calling its internal function that sends a request to the IBM Watson cloud service. After converting the text to speech, the TTS module saves the file to the *TTS Cache Files* folder. The TTS module publishes a message on the *speech* topic (of the Audio module) containing the name of the generated file and ends its execution. The message is received by the Audio module, which calls its internal functions in sequence to read the file from the local folder and play it. When it finishes its task, the Audio module sends an unlock signal by publishing to the special topic, *state*. The message is received by the SP module, causing it to be unlocked, following its flow by executing the next *script* command.

The proposed instantiation of the new EVA 2.0 robot control software encompasses all the functionalities of its extended version. Figure 36 presents a visualization of the complete architecture that was proposed and implemented.

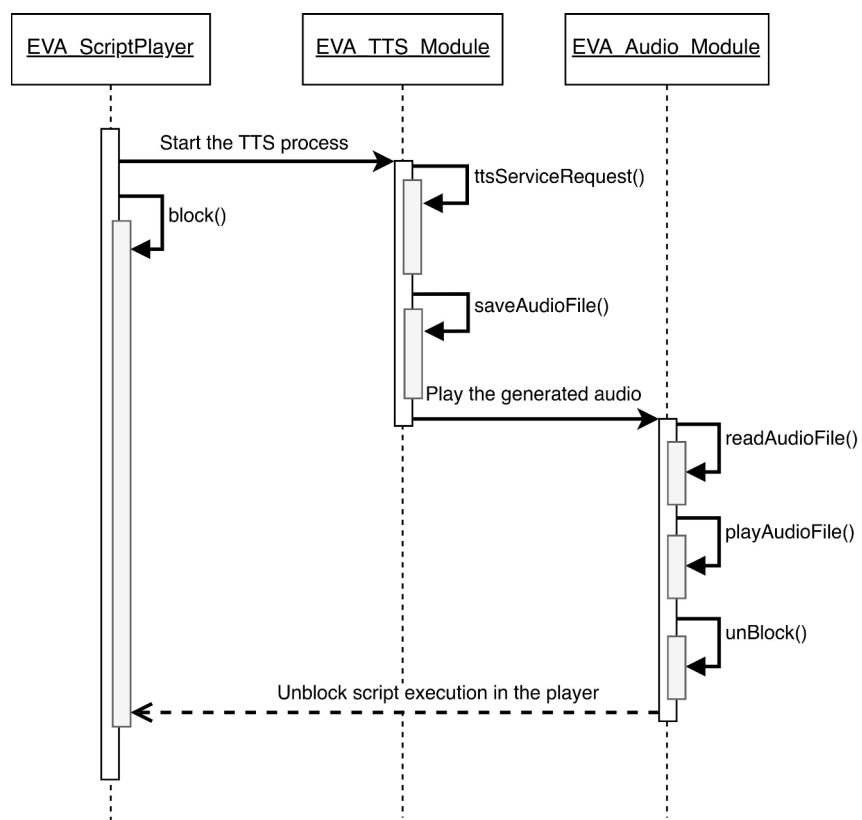


Figure 35: Sequence diagram for the execution of the EVA robot speech feature.

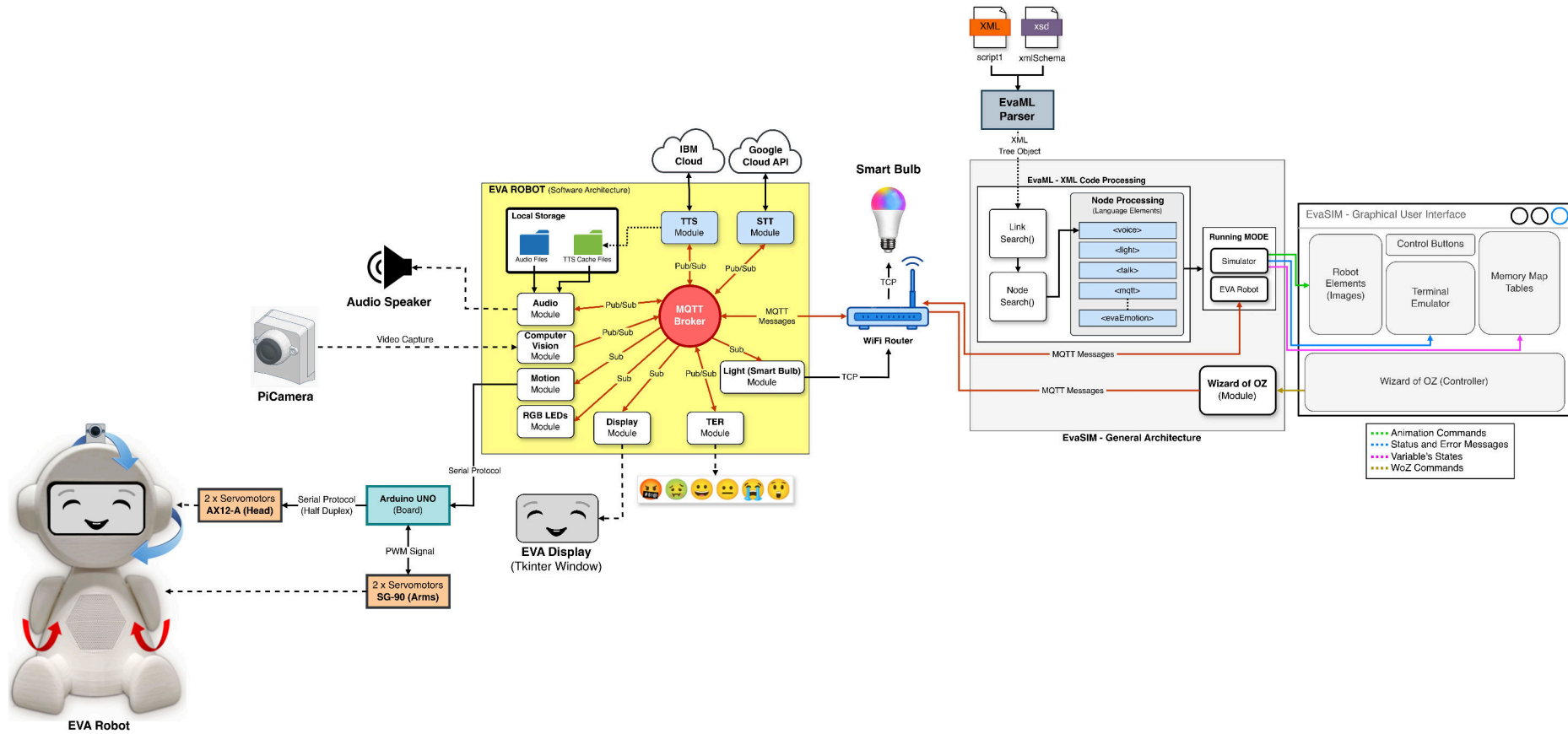


Figure 36: Architecture of the new EVA 2.0 robot software based on the instantiation of the framework proposed in this work.

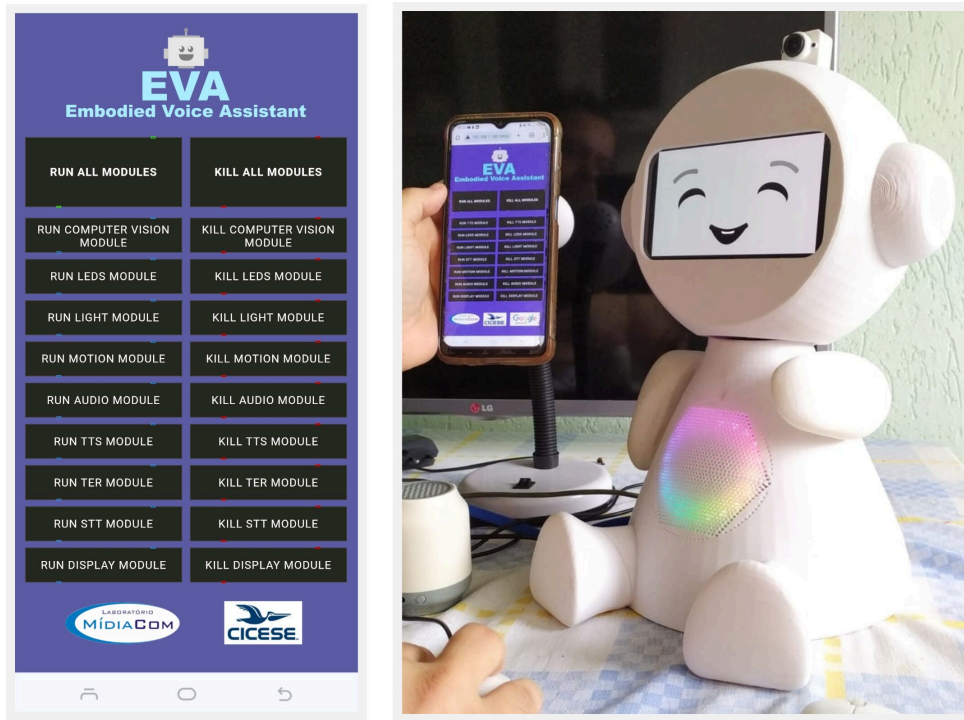


Figure 37: Web application to control the modules of the new EVA software.

In order to complement the EVA system, since the components are independent modules written in Python, a software that controls the execution and termination of each of the modules was implemented. The software runs inside the Raspberry Pi as a web service and allows the user to start and stop each of the modules remotely. The graphical interface of the software running on a smartphone can be seen in Figure 37.

4.2.2 Extending EVA's Nonverbal Communication Capabilities

The modular architecture of the framework enabled the expansion of EVA's non-verbal communication capabilities through the integration of new functionalities. These extensions were designed to enhance the robot's ability to convey emotions via facial expressions and arm movements. The proposed improvements are presented below and include new facial expression designs, featuring an expanded range of emotions, and a new pair of movable arms, enabling coordinated head and arm movements. Together, these additions significantly enhance EVA's expressive capabilities, allowing for more engaging non-verbal interactions.

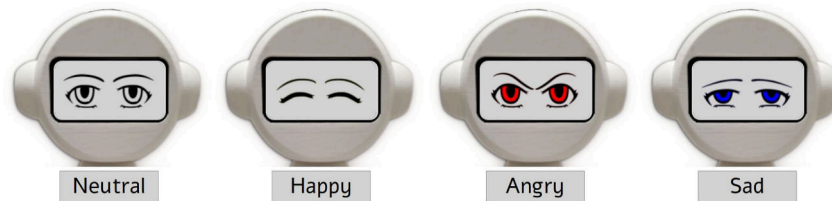


Figure 38: Four expressions available in the original version of the EVA robot.

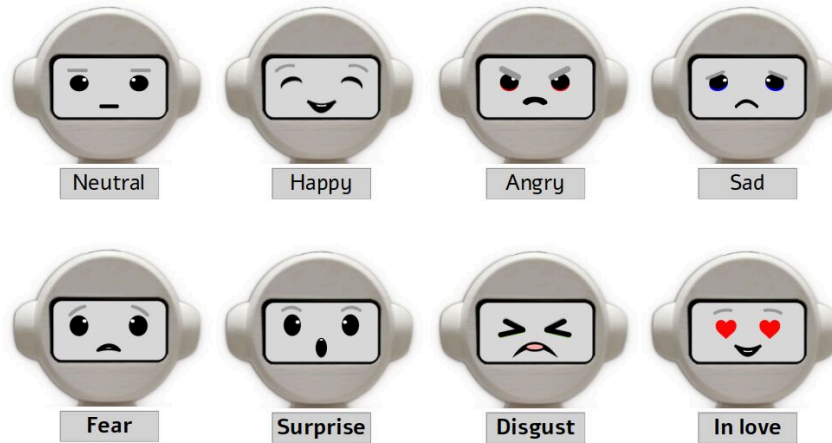


Figure 39: Eight new expressions proposed with a new design for the EVA robot.

4.2.2.1 New Facial Expressions

The EVA robot, in its original version, could express four emotions through its gaze, using its display. These expressions are drawn using only the eyes as elements of the face. These expressions can be seen in Figure 38.

In order to improve the EVA, this work proposes an increase in the number of facial expressions. This work proposes a new design for the robot's expressions using, in addition to the eyes, the mouth, as a graphic element to represent EVA's facial expression. Based on the model of emotions presented in the work of Ekman (1992), 6 expressions are proposed: *anger*, *happiness*, *sadness*, *fear*, *surprise* and *disgust*. In addition to these 6 emotions, there is also the expression of *neutrality*. In addition to these 7 expressions, which are based on real expressions, the expression "*in love*" was added. This is a fun expression, which is intended to be used mainly in robot applications aimed at children. Figure 39 shows the eight proposed expressions.

4.2.2.2 New EVA's Arms

In its original version, the EVA robot features two degrees of freedom (2DoFs), allowing it to move its head both vertically and horizontally. Initially, the robot's arms were

merely attached to the body to complete its anatomical structure, lacking any movement or function. This work proposes adding movement capabilities to the robot's arms. By enhancing its non-verbal communication through the addition of two degrees of freedom to the arm movements, EVA becomes significantly more expressive. EVA can now move its arms up and down (using the UP and DOWN commands), move directly to predefined positions (P0, P1, P2, and P3), and wave its arms around their current position. To enable this arm movement, this work also introduces a new arm design suitable for 3D printing. The available 3D models can be viewed in Appendix B.

4.2.3 Extending EVA's Multimodal Interaction and Affective Capabilities

This section presents the robot's new multimodal interaction capabilities, implemented based on the framework's instantiation. The EVA 2.0 version features a computer vision module capable of performing functions such as facial recognition, QR code reading, and facial expression recognition. The ability to recognize emotions from text has also been added to the robot's system.

4.2.3.1 Computer Vision Module

The multimodal interaction capability of the robot's initial version was enhanced in (ROCHA; VALENTIM, et al., 2022) through the integration of facial expression recognition via a USB webcam. This work proposes replacing the USB camera with a PiCamera. Details regarding the 3D model of the PiCamera mount can be found in Appendix B.

The new computer vision module proposed in this work can be divided into three stages. The first is the video stream capture stage. This stage was developed using the Python module specific to PiCamera. The second stage uses the OpenCV library for face detection and image preprocessing. Stage three contains three submodules that implement and provide the computer vision services. The *TensorFlow*, *Keras*, *OpenCV* and *Face Recognition* libraries were mainly used to implement the submodules. Figure 40 shows the processing pipeline of the EVA Computer Vision module with its three stages.

4.2.3.2 Facial Expression Recognition

The Facial Expression Recognition (FER) submodule is able to infer the user's facial expression presenting in its output a string with one of the seven expressions: *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad* and *surprise*. Facial expression recognition is performed

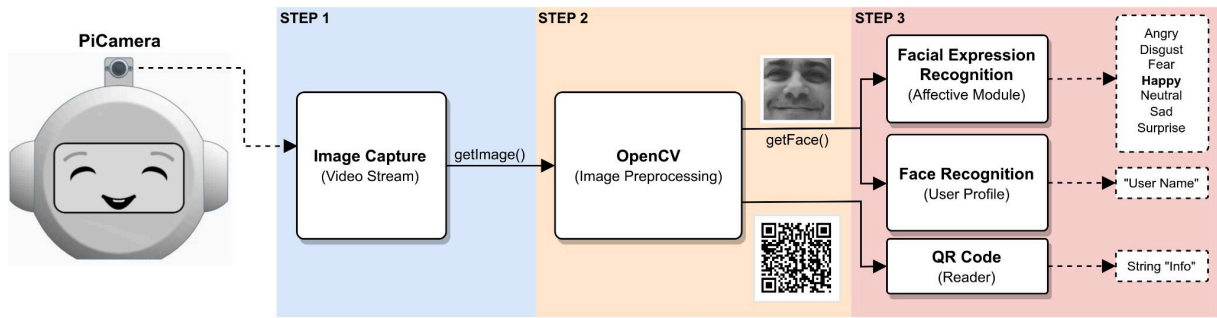


Figure 40: The three steps of the Computer Vision module execution process.

using a convolutional neural network (CNN) based on the implementation provided by Balaji (2020), with a pre-trained model based on the FER-2013 dataset (GOODFELLOW et al., 2013).

4.2.3.3 Facial Recognition

The facial recognition submodule uses a model based on image embedding. Unlike the Eigenfaces model, which requires several photos of the user to train the model to be used in the recognition process, the Encoding model only requires one image of the face. This image is passed to a trained neural network, capable of embedding the face, defining a vector with 128 features capable of identifying it. The comparison between two faces is made by calculating the distances between the two vectors that represent the faces being compared. The user's facial recognition begins with the capture of a frame from the video stream coming from the camera. If this frame contains a face, the frame is processed and the face contained in it is then encoded, resulting in a vector. The distance between this vector and the vectors of the users' faces that were previously encoded and saved in the users folder is calculated. If this distance is within a predetermined value, the comparison is true and indicates that the faces belong to the same person. Figure 41 illustrates how the submodule works.

4.2.3.4 QR Code Reader

This submodule allows a QR Code image to be read, extracting the information (string) contained in it. This functionality allows the robot to be used, for example, in educational activities involving cards (BRAVO PERUCHO; ALIMARDANI, 2023). The reading time of a QR Code is also very fast, depending only on the correct positioning of the image in front of the camera.

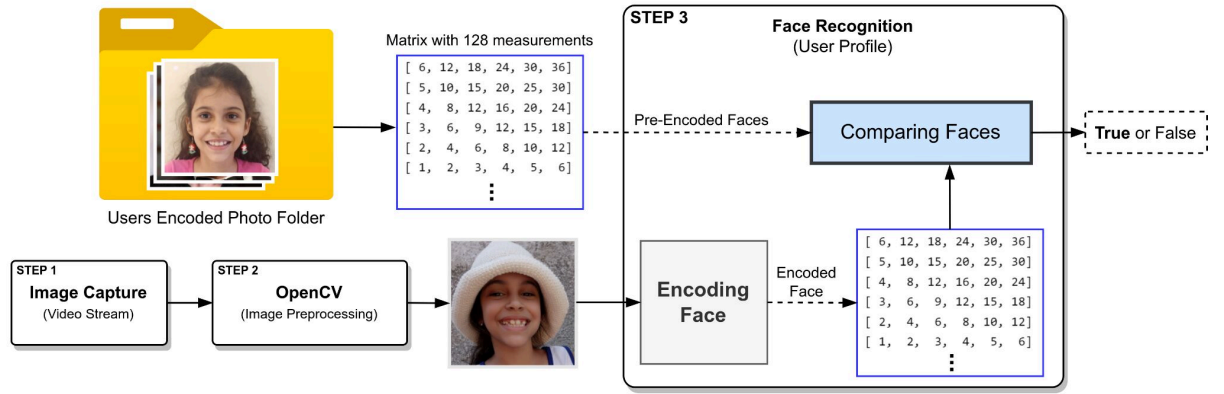


Figure 41: Facial recognition process using the Image Embeddings technique.

4.2.3.5 Text Emotion Recognition (TER)

Text emotion recognition (TER) is an essential task in natural language processing (NLP) research, and its solutions can be useful in a variety of applications, such as information retrieval, learning analytics, data mining, and human-computer interaction (ALSWAIDAN; MENAI, 2020). EVA's 2.0 new TER module uses a large language model (LLM). This model was trained to perform emotion classification tasks in texts. The model can return one of the six emotions: *anger*, *sadness*, *joy*, *fear*, *surprise*, and *disgust*. In addition to the aforementioned emotions, the model can return one more emotion, the *neutral* emotion. The model selected was *emotion-english-distilroBERTa-base* (HARTMANN, 2022).

This section presented the EVA Robot 2.0 instantiation of the proposed framework, including the implementation of its control software and the extensions developed to enhance its nonverbal communication and multimodal interaction capabilities. Table 6 compares the original EVA platform, described in Section A.3, with EVA 2.0 by highlighting the capabilities introduced through the framework instantiation. These capabilities demonstrate the flexibility and extensibility of the proposed framework, illustrating how new functionalities can be incorporated into the robotic system.

The next section presents the instantiation of the framework in the creation and application of the GRACE social robot. The section shows, through the robot project, how researchers were able to extend this version of this EVA-based robot by adding new capabilities to it.

Table 6: Comparison between the original EVA platform and EVA 2.0, highlighting the capabilities introduced through the proposed framework.

Robot Capability	Original EVA	EVA Robot 2.0
Head movement	✓	✓
Smart bulb control	✓	✓
LED colors and animations	✓	✓
Audio play	✓	✓
Text to speech	✓	✓
Speech to text	✓	✓
Facial expression recognition	✓	✓
Number of facial expression	4	8
Integrated camera	—	✓
Face recognition	—	✓
QR code recognition	—	✓
Arm movement	—	✓
Text emotion recognition	—	✓
Distributed modular software architecture	—	✓

4.3 GRACE Robot

GRACE demonstrates the applicability of the framework in a different research context, where it was adopted by researchers at Swiss Federal Institute of Technology in Zurich (ETH Zurich) to support the development of socially assistive robotic applications (VINAY; TOMMILA, et al., 2024; WALLISER et al., 2026). It was developed at ETH Zurich in collaboration with researchers from the University of Zurich, the University of St. Gallen, and Clinic Barmelweid. The project investigated the use of a socially assistive robot to deliver cognitive stimulation activities and conversational interactions aimed at supporting the well-being and quality of life of older adults.

4.3.1 Research Context

The robot was conceived as a derivative version of EVA, reusing both hardware and software concepts provided by the proposed framework. The platform combines speech interaction, visual embodiment, and scripted conversational behaviors to support structured cognitive activities inspired by Cognitive Stimulation Therapy (VINAY; UETOVA, et al., 2025). The implemented activities included memory exercises, guided conversations, and relaxation sessions designed to encourage engagement and social interaction.

4.3.2 Framework Use and Extensions

From a software perspective, GRACE reused several components provided by the proposed framework. The project adopted the same MQTT-based communication architecture originally employed by EVA, in which interaction scripts are executed through EvaSIM 2.0 and translated into commands delivered to the robot through a message broker. The XML-based scripting language (EvaML 2.0) was also used to define conversational flows, intervention activities, and interactive sessions. This allowed the researchers to implement the application using the existing scripting and execution infrastructure without modifying the underlying communication mechanisms. Listing 4.2 presents an excerpt from an interaction script used with the GRACE robot, employing the `<generate_answer>` command, an EvaML extension added by the researchers, with the prompt sent via the `instruction` attribute.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <evaml
3   name="SessionGrace_german"
4   xsi:noNamespaceSchemaLocation="evamlschema/evaml_schema.xsd"
5   <settings>
6     <voice tone="German" />
7   </settings>
8   <script>
9     <evaEmotion emotion="HAPPY" />
10    <talk>Hallo! Wie heissen Sie?</talk>
11    <listen />
12    <!-- New EvaML command added by the researchers -->
13    <generate_answer context="" instruction="Extract the name from the
        user reply. Return only the name. For my name is John reply John
        if there is no name, just return an empty string" />
14    <talk>Hallo $! Es ist schön, Sie kennenzulernen!
        Mein Name ist Grace und ich bin Ihre Sprachassistentin.
15    </talk>
16    <talk>Ich bin hier, um heute mit Ihnen einige Übungen zu machen
        Aber zuerst sollten wir uns gegenseitig kennen lernen!
17    </talk>
18    <listen />
19    .
20    .
21  </script>
22 </evaml>

```

Listing 4.2: Excerpt from an interaction script used with the GRACE robot using the proposed framework with the EvaML language extended to meet the research requirement.

In addition to reusing existing framework components, the project introduced several

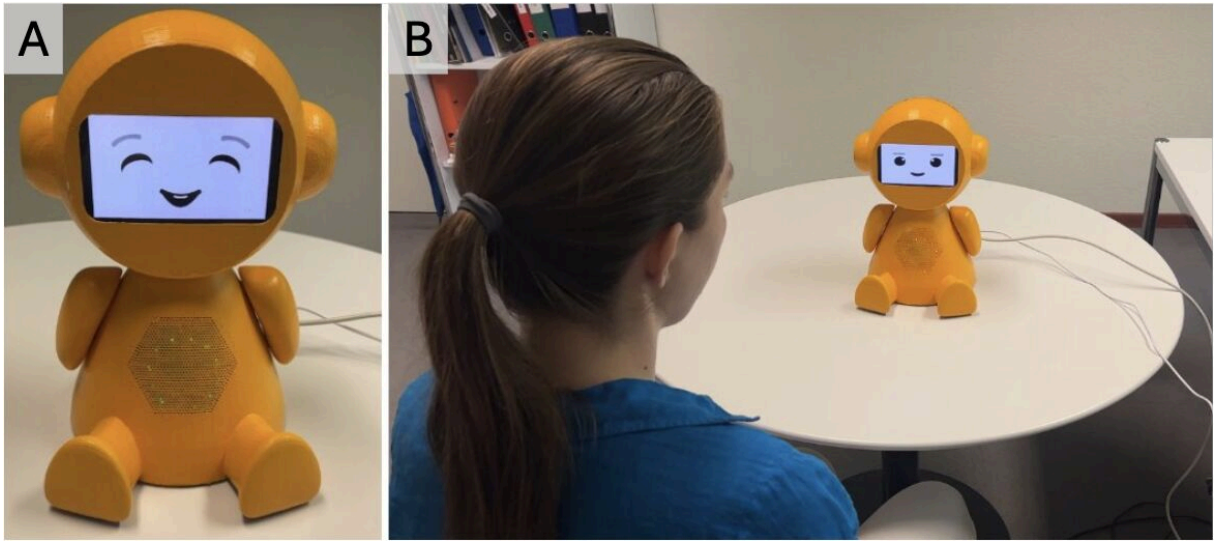


Figure 42: GRACE robot. (A) Front-facing image and (B) with a test user (VINAY; UETOVA, et al., 2025).

extensions to meet the target application’s requirements. Consequently, EvaML 2.0 was extended to interact with the OpenAI API, enabling the robot to generate contextual responses based on user input while preserving the structure of rule-based interaction scripts. Furthermore, the robot’s control modules were extended to enhance its voice and expressiveness capabilities through the integration of the ElevenLabs text-to-speech service, resulting in more natural and consistent voice output. Additionally, during the robot’s assembly, a hardware component was discontinued and had to be replaced with a similar model from another manufacturer. The framework’s modular design allowed the audio processing module to be easily adapted to the new hardware control module.

The GRACE project, presented in this section, illustrates the instantiation of the framework in a new robot based on EVA, involving the reuse of the communication infrastructure, the use of programming, testing, simulation, and control support tools (EvaML 2.0 and EvaSIM), and the extension of the framework through the introduction of new functionalities required by the target application. This provides evidence that the proposed framework enables HRI researchers to build and/or extend robotic platforms for use in various research contexts. Figure 42 shows an image of the GRACE robot.

The following section presents the instantiation of the framework in the construction and application of the FRED MX robot within an educational context at an institution in Mexico.

4.4 FRED MX Robot

FRED MX is a version of the affective social robotics platform based on the FRED robot, which will be presented in the next chapter. FRED MX represents an instantiation of the framework proposed in this thesis in a research project conducted at the Center for Scientific Research and Higher Education of Ensenada (CICESE) in Mexico. The platform was employed in a research project investigating the use of affective conversational strategies to support emotion regulation and mitigate academic procrastination among adolescents.

4.4.1 Research Context

The project explored how verbal and non-verbal affective behaviors could influence students' perceptions of emotional support in educational settings. A systematic process was conducted to identify evidence-based strategies for procrastination management, which were subsequently translated into robot interaction scenarios. These scenarios incorporated planning assistance, goal setting, task organization, motivational support, and emotion regulation strategies. The resulting system was evaluated through a study involving high-school students, comparing affective and neutral interaction conditions.

4.4.2 Framework Use and Extensions

FRED MX reused the software architecture and robotic functionalities provided by the proposed framework. The system employed the same publish-subscribe communication model based on MQTT, the distributed architecture composed of low-level and high-level processing layers, and the existing control infrastructure originally developed for the FRED platform. The low-level layer remained responsible for controlling actuators, sensors, and expressive devices through the Arduino UNO and ESP8266 boards, while high-level functionalities executed on an external computer.

The project also reused a variety of existing robot capabilities, including facial expressions, body movements, lighting effects, and conversational behaviors. These functionalities were combined to create interaction scenarios designed to support emotion regulation and study planning activities. Unlike the GRACE project, presented in Section 4.3, which required extensions to the scripting infrastructure and integration with external AI services, FRED MX mainly relied on the adaptation and orchestration of

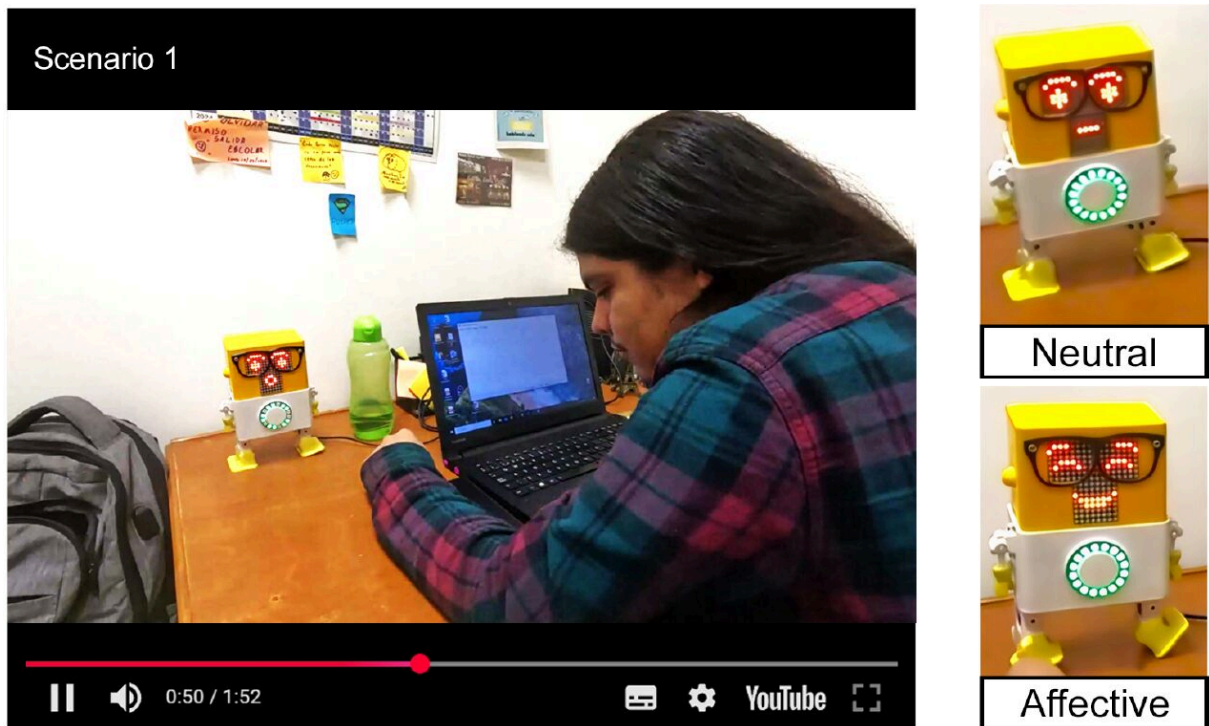


Figure 43: FRED MX robot. Left: setup shown in the videos used for both experimental scenarios. Right: the two robot conditions – neutral behavior (top) and affective behavior (bottom) (RANGEL et al., 2025).

functionalities already available within the framework. By relying on existing framework components, the researchers were able to focus on interaction design and experimental evaluation rather than on the implementation of low-level robotic functionalities.

Beyond its application domain, FRED MX represents an instantiation of the proposed framework in the context of affective educational robotics. The project demonstrates how the framework can be employed to support the development of domain-specific interaction strategies while reusing existing robotic functionalities and software infrastructure. Figure 43 shows some images of the FRED MX robot as well as the scenario used in the experiments.

In addition to the framework instantiations presented in the previous sections, individual components of the proposed framework have also been independently reused in other projects. The following examples illustrate this type of reuse through applications of SENSEI.

4.5 Reuse of Framework Components

In addition to the framework instantiations presented in the previous sections, one of its components was also independently reused to build two applications. This section presents two projects developed by undergraduate students that utilize SENSEI, a component of the Execution Layer, as a script player for executing EvaML scripts.

4.5.1 EvaPlay

EvaPlay¹ is a web-based authoring environment designed specifically to unify and simplify the cycle of creating, simulating, and refining interactive scripts for the socially assistive robot EVA. It enables professionals without a background in computing, such as therapists, educators, and caregivers, to independently customize interactive sessions, facilitating games, dialogues, or emotional interventions tailored to each user's profile. Consequently, the environment significantly reduces the cognitive load associated with technical learning and programming syntax, offering a space where authors can focus exclusively on interaction logic and the expected behavior of the social robot.

Figure 44 shows EvaPlay's terminal emulation running an EvaML script by reusing the framework's SENSEI component (Execution Layer). Just as in the terminal mode provided by SENSEI, various important details are displayed, such as the countdown for `<wait>` commands, the states and colors of the smart bulb (triggered by the `<light>` command), and emojis representing the robot's facial expressions activated by the `<evaEmotion>` command.

4.5.2 EVA Virtual

EVA Virtual is a mobile application designed to faithfully replicate the interactive functionalities and multimodal communication logic of the EVA robot (MOREIRA et al., 2026). The primary purpose of this tool is to enable the continuity of treatment and therapy for children with Autism Spectrum Disorder (ASD) by offering an accessible, low-cost alternative that runs directly on widely available devices, such as smartphones and tablets.

Structurally, SENSEI is encapsulated in a container alongside a mini-API dedicated to communicating with the orchestrator, which acts as the main server. The communication

¹<https://eva-script.onrender.com/>

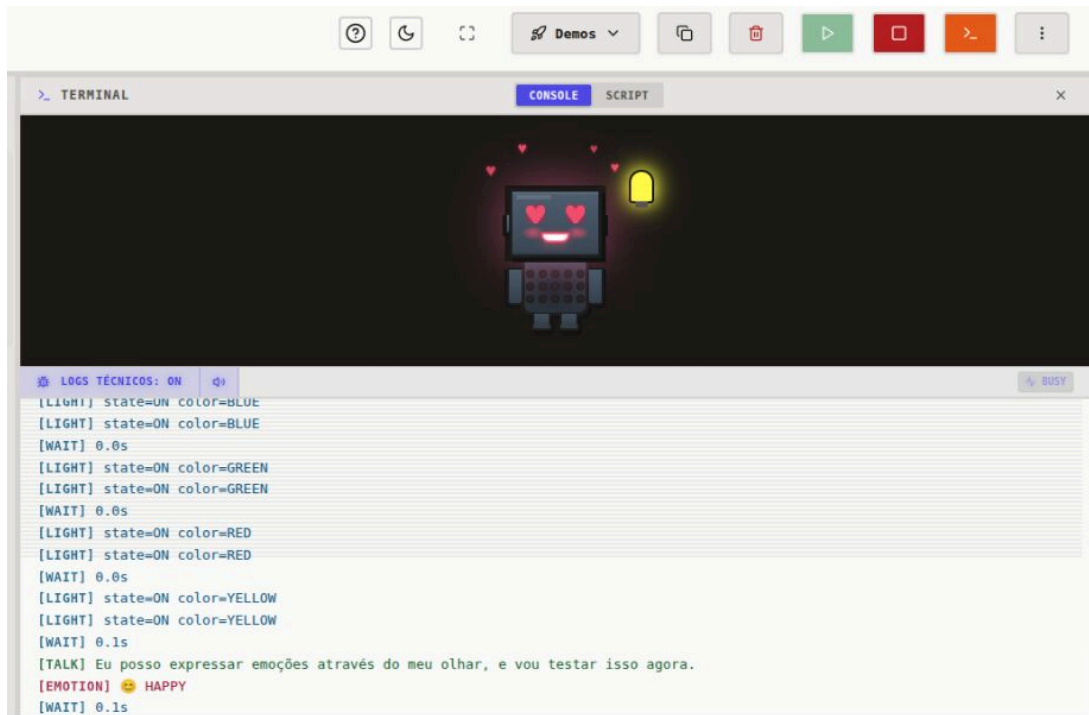


Figure 44: Terminal emulation of the EvaPlay software. A case of reusing the SENSEI component (Execution Layer).

flow occurs when SENSEI sends a command via MQTT, a second container reads this topic and translates it into a channel monitored by the system’s internal API, which then converts the message into JSON format and sends it via WebSocket to the orchestrator for processing and final adjustments. Figure 45 shows the Virtual EVA executing a `<talk>` and `<evaEmotion>` commands of the EvaML language with the aid of reusing the SENSEI component from the Execution Layer of the framework.

4.6 Final Remarks

This chapter presented different instantiations of the proposed framework, illustrating its application at various levels. First, the Affective Layer was instantiated through concrete implementations of the extension points defined for PULSE, ROSE, and AEB. The resulting components were integrated into the Execution Layer and demonstrated in two affective interaction scenarios using an affective robotics platform. The chapter also presented the instantiation of the framework with version 2.0 of EVA, as well as the GRACE and FRED MX, robotic platforms developed using elements derived from the architecture and abstractions provided by the framework. Furthermore, two applications, EvaPlay and EVA Virtual, demonstrate the feasibility of creating additional applications



Figure 45: Image of the Virtual EVA running on a smartphone ([MOREIRA et al., 2026](#)). Another case of reusing the SENSEI component (Execution Layer).

by reusing framework components, such as SENSEI (Execution Layer). To complete the framework instantiation cases, one more instantiation will be performed on the FRED robot in Chapter 6.

In summary, these instantiations and instances of reuse demonstrate that the proposed framework can support different robotic platforms across diverse research contexts and interaction scenarios, while simultaneously allowing for implementation choices tailored to specific requirements.

Each new feature added to the robot, whether associated with hardware or a new software capability, directly impacts the robot's language and the simulation software, EvaSIM. The next chapter presents the contributions of this thesis that are directly related to extensions of the language and the simulator.

5 Extensions of EvaML and EvaSIM

EvaML and EvaSIM ([ROCHA; FAVELA; MUCHALUAT-SAADE, 2025](#); [ROCHA; CRUZ-SANDOVAL, et al., 2024](#)) are development tools associated with the proposed framework. Although they are not architectural components of the framework, these tools complement its capabilities by providing support for script development using a domain-specific language, as well as for testing, simulation, and the execution of interaction scenarios. Both tools were proposed in ([ROCHA, 2022](#)), and in this thesis, they were extended to support the new capabilities introduced by the proposed framework.

Many features related to the framework were presented in previous chapters. Upon receiving a "START" message via a command in the robot's script, PULSE can accept multimodal inputs once activated by the opening of an Affective Perception Window. EVA version 2.0 introduced a range of new features, such as QR code reading, text-based emotion recognition, and new head and arm movement types. Each of these additions required extending the EvaML language to support the new capabilities through the introduction of new XML elements (commands). Implementing these changes involves the following steps: 1) defining the command name and attributes in an .xsd file (if applicable); and 2) implementing the Python module that is dynamically loaded during script execution. As seen in Section [A.5](#), the EvaSIM simulator was originally capable only of running EvaML code within a graphical interface, executing the script on the physical robot required sending the code generated by the EvaML parser to the robot to be run by its native executor.

This chapter presents the new EvaML 2.0 commands and the new EvaSIM 2.0, which was rebuilt based on the framework's elements and extended to allow these new EvaML commands to be executed directly on the physical robot. This chapter is organized as follows. Section [5.1](#) presents the extensions to the EvaML language, resulting in EvaML 2.0, while Section [5.2](#) covers the extensions to the EvaSIM simulator, which has been renamed EvaSIM 2.0.

5.1 EvaML 2.0

With the addition of new capabilities to the EVA robot, it was necessary to update its programming language, EvaML (ROCHA; FAVELA; MUCHALUAT-SAADE, 2025). This was done by creating new elements and attributes. Since EvaML is an XML-based language, these new elements and attributes were added to the XML Schema file responsible for specifying the language grammar and can be seen in Table 7. The elements and attributes highlighted in blue are part of the new additions made to the EvaML 2.0 language.

As shown in the Table 7, EvaML 2.0 comprises 34 elements. Regarding the elements highlighted in blue, 13 new elements were created, and 6 new attributes were added to elements that already existed in the previous version of the language. In total, the new 2.0 version features nearly twice as many elements. This section briefly describes these elements and attributes.

The `<loop>` element. In order to facilitate the creation of repetition loops, this work proposes the addition of the `<loop>` element to the EvaML language. The `<loop>` element executes "n" times the block of commands specified inside it. By using the `var` attribute it is possible to define the variable that will be used in the iterative process (the loop control variable) and this variable can be accessed in the code. If no variable is defined by the user, the parser will generate one automatically. As it determines the number of repetitions the loop will execute, the `times` attribute is mandatory. For its value, positive values or other variables previously defined by the user are accepted.

The new `<var>` attribute. A significant conceptual addition to the EvaML language was the ability for the developer to define the variable that receives values returned by commands generating storable information, such as `<random>`, `<listen>`, `<userEmotion>`, etc. In the first version of EvaML, values returned by these elements were stored arbitrarily in memory and could only be referenced using the "\$" indexing symbol. Now, thanks to the `var` attribute, the returned or generated string can be stored in the variable specified within the command itself.

The `<talk>` and voice elements. In addition to the ability to define the robot's voice at the script level within the `<settings>` section, it is now possible to change the robot's voice locally using the `voiceType` attribute. This change is local and temporary, applying

Table 7: Elements and attributes of the EvaML language. Those highlighted in blue result from the extension of the EvaML language, which is one of the contributions of this thesis.

Functional Category	Element	Attribute(s)
Control Flow	switch	<u>id</u> , var
	case	<u>op</u> , <u>value</u>
	default	
	loop	<u>id</u>, var, times
	goto	<u>target</u>
	wait	<u>id</u> , <u>duration</u>
	stop	
Numeric Operation	counter	<u>id</u> , var, <u>op</u> , value
	random	<u>id</u> , <u>min</u> , <u>max</u> , var
Verbal Communication	talk	<u>id</u> , voiceType
	voice	<u>type</u>
Audio	audioEffects	<u>mode</u> , vol
	audio	<u>id</u> , <u>source</u> , <u>block</u>
Nonverbal Communication	evaEmotion	<u>id</u> , <u>emotion</u>
	led	<u>id</u> , <u>animation</u>
	lightEffects	<u>mode</u>
	light	<u>id</u> , <u>state</u> , color
	motion	<u>id</u> , <u>type</u> , head, leftArm, rightArm
	pose	<u>id</u>, <u>type</u>
Multimodal Interaction	listen	<u>id</u> , var, language
	userEmotion	<u>id</u> , var
	textEmotion	<u>id</u>, var, language
	qrRead	<u>id</u>, var
	userID	<u>id</u>, var
	perception	<u>id</u>, <u>action</u>, sources
Affective Configuration	robotAffectiveProfile	empathy, decay, decayDelay, baseMood, profile
LLM Integration	llm	<u>id</u> , var
	llmRobotProfile	temperature, numPredict
Network Communication	http	<u>id</u>, url
	mqtt	<u>id</u>, pubTopic
Script Organization	macro	<u>id</u>
	useMacro	<u>macro</u>
Utility	debug	<u>id</u>, <u>operation</u>
	log	<u>id</u>, <u>name</u>

only to the command currently being used.

The `<motion>` element. The new version of the `<motion>` command now utilizes EVA's new head movements and can also control the robot's arm movements. The `<pose>` command shares similar semantics but applies to a different robotic platform capable of various poses: FRED.

The `<listen>` element. The `<listen>` command can specify the variable that will receive the string from the STT service using the `var` parameter, and also define the capture language through the `language` attribute.

The `<textEmotion>` element. The `<textEmotion>` element is responsible for enabling EvaML to recognize emotions in text. The `language` attribute specifies the language of the text to be classified. If the text is in English, no translation is required, and classification takes place immediately (locally); if it is in another language, the text is translated via the Google API and then forwarded to the emotion classification stage.

The `<qrRead>` and `<userID>` elements. These two elements share the use of the camera to obtain information. The first reads a string contained in a QR code, while the second is capable of identifying a user registered in the system by capturing an image of their face. Both use the `var` attribute to define where the information will be stored.

The `<perception>` element. It is one of the key elements introduced based on the framework's affective components proposal. It initiates, configures, and ends an Affective Perception Window. Its usage and attributes have already been detailed in Section 4.1.4.2, within the discussion on interactive affective scenarios.

The `<robotAffectiveProfile>` element. This element is responsible for configuring the robot's affective profile. It must be used within the `<settings>` section and is executed when the script begins loading. Its `empathy`, `decay`, `decayDelay`, `baseMood`, and `profile` attributes allow for the configuration of the parameters that define the desired profile.

The `<llm>` and `<llmRobotProfile>` elements. These elements enable EvaML to utilize a local LLM through integration with Ollama (OLLAMA INC., 2026). The `<llmRobotProfile>`

must be used within the `<settings>` section. It executes at the very beginning of the script, loading the specified model and providing the initial system prompt for the model to process, it also allows for the configuration of various model behaviors. The `<llm>` element allows sending text with prompts that can contain user-defined variables as well as the "\$"¹ variable. The LLM's string output can be assigned to "\$" or to a variable defined in `var`.

The `<http>` and `<mqtt>` elements. As the names suggest, these elements allow messages to be sent using the respective protocol.

The `<debug>` and `<log>` elements. The `<debug>` element was created to assist the script developer during the debugging process. It can perform three functions: pause the script and wait for a key press; print script-related variables (such as "\$" values and user-defined variables) to the screen; and print system variables, primarily those related to data sent by components of the framework's Affective Layer. The `<log>` sends the messages contained within its body via MQTT. These messages can include script variables and are sent to the "LOG" subtopic.

5.2 EvaSIM 2.0

As presented in Section A.5, the EvaSIM simulator (ROCHA; CRUZ-SANDOVAL, et al., 2024) was developed to assist in the learning and development of EvaML scripts for the EVA robot. One of its applications is to serve as a testing and debugging platform for scripts developed for the physical robot, representing, in a simplified manner, the physical elements that make up the robot, such as: the Matrix Voice board, the display with the robot's expressions, the smart bulb, etc.

Based on the proposed modular, distributed architecture based on the MQTT communication protocol presented in the proposed framework, this work also proposes a new version for the simulator, the EvaSIM 2.0. In this new version, the simulator now has two running modes: Simulator Mode and EVA Robot Mode. In addition, a Wizard of OZ (WoZ) control panel was added to its graphical user interface, which allows the robot to be operated remotely through buttons that control its functionalities. Figure 46 presents the

¹If no user variable is defined in the `var` attribute of an input element (such as `<listen>`, `<userEmotion>`, etc.), the input result is placed at the end of a vector-like structure in the robot's memory. The "\$" character points to the last element of this vector and can be referenced within the EvaML script.

new EvaSIM interface with all its new elements and features. In the figure, four regions are highlighted. Each of these regions and their respective elements will be described below.

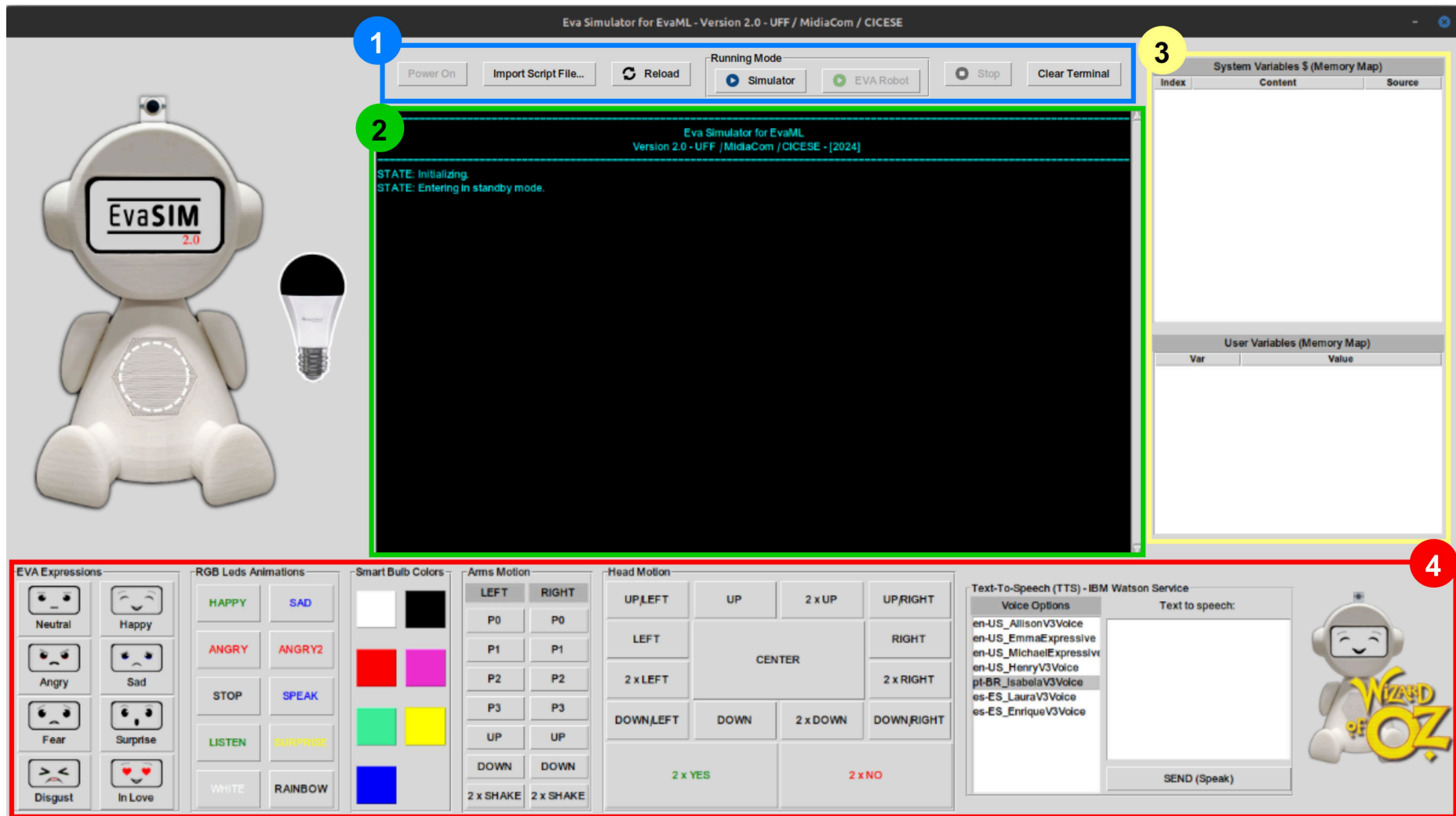


Figure 46: New EvaSIM 2.0 simulator interface and its regions. (1) EvaSIM control bar. (2) Terminal emulator. (3) Tables with the simulator's memory maps. (4) Wizard of OZ interface controls.

5.2.1 EvaSIM - Graphical User Interface

This section describes the elements of the EvaSIM simulator's graphical user interface that have been improved since the previous version. To facilitate visualization of the elements, the interface has been divided into regions.

EvaSIM Control Bar. Region (1) is the simulator's control bar, which now features new buttons and functionalities. The "Reload" button allows you to reload a script file already loaded into the simulator's memory. Since the simulator now functions as a player for EvaML scripts, causing script commands to act directly upon the physical robot, it offers two execution modes. These modes can be selected using the new buttons: the "Simulator" button, which executes the script using the simulator's graphical interface elements, and the "EVA Robot" button, which executes the script commands by controlling the physical robot.

Running Modes. As previously mentioned, the simulator can now operate in "Simulator" and "EVA Robot" modes. In the first mode, the simulator functions as it did in the previous version, simulating the behavior of the physical robot, simulating voice capture via a window with a text box, and using the keyboard as the input interface. To simulate facial expression recognition, EvaSIM uses a window with expression options represented by emojis. In "EVA Robot" mode, the simulator acts, in part, as a script executor for the physical robot. Script elements controlling the robot's components act directly on the hardware. Furthermore, the robot's speech is controlled by the physical robot's text-to-speech module. It is important to note that, even in "EVA Robot" mode, the simulator continues to display the executing elements in its terminal, and variables remain visible in the graphical interface's memory tables. In this mode, voice capture is performed by the physical robot's speech-to-text module.

Terminal Emulator. The EvaSIM terminal emulator can be seen in region (2). In this new version, in addition to having its data presentation area expanded by increasing the number of lines and columns, a scroll bar has been added to the right side of the terminal, allowing greater control over the display of text that exceeds its limit on the number of visible lines.

Memory Map Tables. Region (3) contains two tables. They display the contents of the variables created in the EvaML script. They can be created in two ways: they can be defined statically by the script developer, or they can be created dynamically from one of the robot's multimodal inputs, being stored in the special variable "\$".

5.2.2 EvaSIM - Wizard of Oz

The last region to be discussed is the one that contains the controls for the robot's Wizard of Oz (WoZ) mode. WoZ is one of the most popular techniques for conducting initial interactions to design and define future autonomous interactions for the robot. Region (4) of the EvaSIM GUI provides a series of elements that allow you to create WoZ interactions using predefined expressions, smart bulb colors, RGB LED animations, arm and head movements, and phrases that can be written for the robot to speak.

EVA Expressions. Each of the buttons in this group represents a facial expression of the robot and allows the user to change the EVA's facial expression to one of its eight available expressions.

RGB LEDs Animations. The buttons in this group can execute one of the 10 possible LED animations available for the EVA. As you can see in the interface, colors are associated with some emotions.

Smart Bulb. The smart bulb's button group allows it to be turned on and off as well as having its color selected through the colors available in the interface.

Arm Motion. In this new proposal for the EVA robot, the ability to move its arms has been added. The "Arm Motion" group shows 14 buttons (seven for each arm) with the four possible types of movements (P0, P1, P2, P3). It is also possible to raise and lower the arms using the "UP" and "Down" commands, and it is also possible to make the arms shake up and down, around their current position.

Head Motion. The set of buttons controlling the robot's head movement represents a subset of the movements possible for EVA's head.

Text-To-Speech (TTS). The last group of widgets in region (5), the WoZ region, allows the user to listen to audio generated by IBM Watson from texts typed into the interface through the robot. This feature can be used to check how the robot speaks when using certain phonemes.

5.3 Final Remarks

This chapter presented EvaML 2.0 and EvaSIM 2.0, two new versions of the language and the simulator that emerged from the contributions of these theses regarding extensions to both. EvaML was extended to support over a dozen new features linked to the EVA robot 2.0 and the framework's Affective Layer components. Its multimodal interaction capabilities were expanded, and it can now utilize the resources of a local LLM. In addition to its existing role as a simulator, the new version of EvaSIM (2.0) can execute scripts that interact directly with the physical robot's hardware, it sends commands and transmits or receives parameters and responses resulting from user interactions, such as a text string from a voice interaction or a QR code scan. This work also proposed adding new elements (buttons) to the EvaSIM graphical interface, allowing the user to remotely control various robot functions (Wizard-of-Oz setup).

The next chapter presents another main contribution of this thesis: a low-cost open-source affective social robotics platform called FRED. It is a robot designed for applications in healthcare and education, and can be used by researchers in experiments involving affective human-robot interaction. It was used in the interactive scenario demonstration video, in Section 4.1.4. Furthermore, the chapter presents an evaluation of the platform's affective expression capabilities and robotic characteristics.

6 FRED Affective Social Robot

This work has proposed several extensions for the EVA robot, significantly expanding its multimodal interaction capabilities. However, building a new robot from scratch by modeling its 3D elements and defining its hardware elements offers important benefits in terms of customization and control, as it allows adapting the robot's functionalities to specific applications and needs, enhancing its interaction capabilities and ensuring that it meets the desired specifications for various environments.

Despite being an open-source social robotics platform, EVA uses materials that are still expensive, especially when compared to the reality in Brazil. According to (CAO et al., 2014), the great challenge to expanding the scope of the use of social robots and applying their results in people's daily lives is the high cost of these platforms.

This thesis proposes a low-cost, open-source affective social robotics platform called FRED (Friendly Robot for Education and Healthcare) (ROCHA; MUCHALUAT-SAADE, 2023; RANGEL et al., 2025). The platform utilizes a flexible software architecture based on the framework proposed in Chapter 3. FRED possesses several features that distinguish it from EVA and justify its creation, construction, and use in this thesis. Some of its functionalities have been inspired by related work, as follows.

Leg movements in robots can enhance their emotional expression and social interaction. According to Garcia, Gomez-Donoso, and Cazorla (2024), integrating leg movements with other multimodal elements, such as visual and auditory cues, improves emotion recognition. Ali et al. (2023) state that varying speed, frequency, and joint angles, robots can convey emotions with an accuracy of 79.69%, demonstrating the importance of dynamic body gestures in emotional expression. A study involving the Yanshee robot demonstrated that humans could perceive emotional expressions through robot body movements, suggesting that leg movements can be designed to convey specific emotions like happiness, anger, and sadness (GAO et al., 2024). According to Darda et al. (2024), the dance of a social robot can influence human perception of its likability, as it involves various factors

such as the robot’s design, movement, and the context in which it is perceived. The perception of likability is closely tied to how human-like the robot appears and behaves.

According to [Laeque and Akmal \(2017\)](#), colors can increase children’s motivation for learning, and colorful environments and materials in the classroom can increase engagement, making learning more enjoyable and effective. In addition, the emotional design hypothesis suggests that the use of colors in learning materials can provoke positive emotions, guide attention, and promote learning ([DÉSIRON; SCHNEIDER, 2024](#)).

[Ashok, Paplu, and Berns \(2023\)](#) state that a robot’s design, including its costume and physical characteristics, can improve its likeability. For example, robots dressed in casual attire are often perceived as friendlier and more approachable, which may increase their likeability in social contexts.

FRED follows design standards aligned with the study presented by [Rudenko et al. \(2024\)](#). According to the authors, the robot’s aesthetic design and form should not convey a highly specific, rigid, or fixed visual identity, instead, they should leave room for children to use their imaginations and invent their own scenarios and games. FRED wears a very charming gray scarf. It is a multicolored robot with a simplified design that does not aim to resemble the human form. According to [Cano et al. \(2026\)](#), children naturally tend to anthropomorphize robots, attributing human characteristics to them and forming emotional bonds. The authors suggest that an appearance fostering this connection, without striving for physical realism, helps build trust and empathy within the learning environment.

FRED is capable of speech synthesis (TTS) and speech recognition (STT). It can also control the effects and colors of the LED ring on its chest. Furthermore, FRED can express emotions via a simple display made of three 8x8 LED matrices, and it can move using its legs, strike poses, and even dance. [Cano et al. \(2026\)](#) state that communicating readable affective states (such as joy, sadness, and surprise) does not require realistic faces or complex expressions, the robot can express these emotions highly effectively through simple multimodal cues involving head movements, colored LED lights, and sounds.

This chapter is organized as follows. Section [6.1](#) presents the proposal for the FRED affective social robotic platform. Section [6.2](#) describes the robot control software as a concrete instantiation of the proposed framework. An evaluation of FRED’s affective expression capabilities and robotic characteristics is presented in Section [6.3](#).

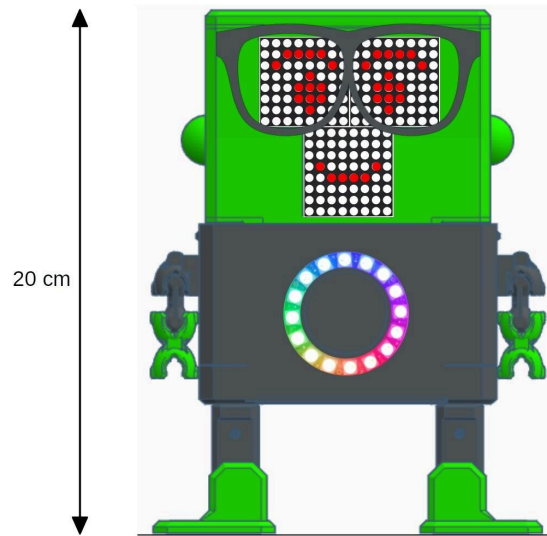


Figure 47: FRED robot with its proper proportions and height in centimeters.

6.1 Friendly Robot for EDucation and Healthcare: FRED

The creation of FRED was inspired by two platforms. The first one (the Otto robot¹) served as the basis for the models of its body parts and for the 3D structure of FRED's legs. The second one, the EVA² robot, served as inspiration for adding elements of verbal and non-verbal communication to the robot, such as the ability to express emotions through its gaze and the use of color elements with the use of RGB LEDs on its thorax. FRED has two arms that are articulated, but do not have any automatic control. The robot is about 20 centimeters tall and approximately 182 grams of PLA filament are used to print all of its parts. Fully assembled, with all hardware components, FRED weighs about 345 grams. Its approximate cost is USD \$70 (quote made on March 20, 2025). Figure 47 shows a 2D projection of the FRED robot with its proper proportions and height in centimeters.

6.1.1 Models for 3D Printing

The platform TinkerCad³ was used to create and edit the 3D models. Figure 48 shows the parts of the FRED 3D model in TinkerCad. The description of each of these parts can be seen in the following items: (1) is the 3D piece that represents FRED's head; (2) its thorax with holes for the screws and a socket to receive the RGB LED ring; (3) is the piece that represents the robot's shoulder and is screwed to the thorax; (4) and (5) are

¹<https://www.ottodiy.com/>

²<https://eva-social-robot.github.io/>

³<https://www.tinkercad.com/>

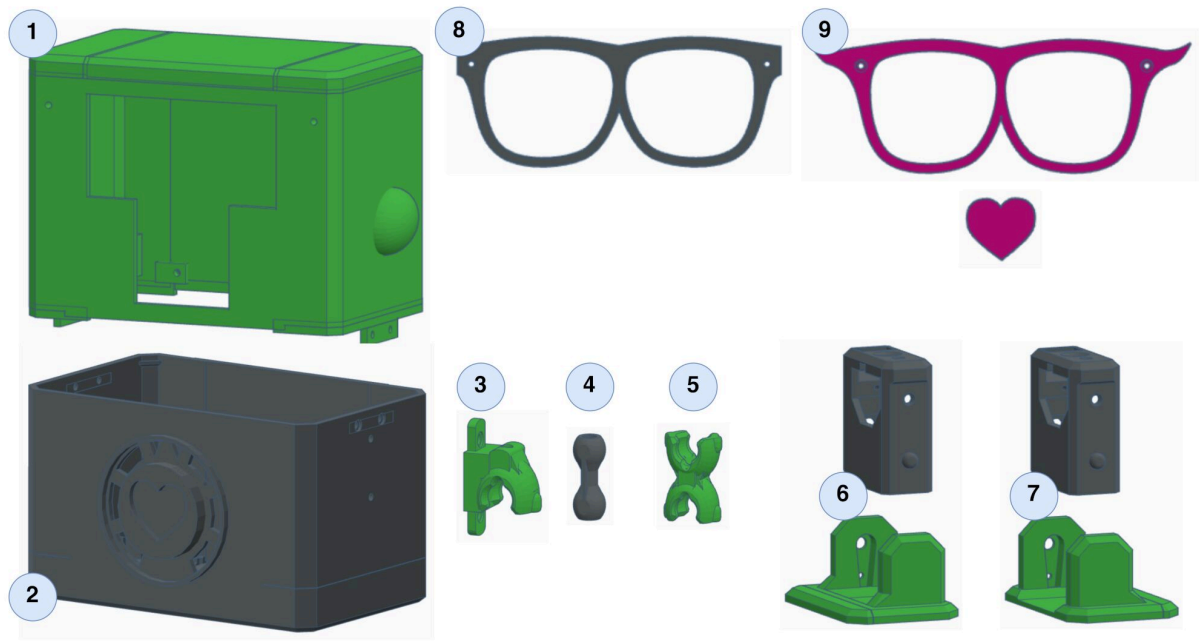


Figure 48: 3D models of FRED and some customized elements for the assembly of his female version, Frida, on the TinkerCad platform.

the articulated arm and hand pieces; (6) and (7) are the legs and each of the robot's feet; (8) is the 3D model of FRED's glasses; (9) it is also possible to see some customization elements of the robot, such as the glasses and the heart that is attached to its chest, in the female version of FRED, the Frida robot. These models can be accessed through this link⁴.

The STL versions for each of the FRED parts can be downloaded directly from its repository through the link⁵.

6.1.2 Hardware Components

FRED is an open-source robotics platform that serves as a model for an affective social robot capable of expressing emotion through non-verbal communication elements using low-cost components, enabling its adoption in educational and healthcare settings in developing countries. Therefore, the FRED hardware design aims to use easily acquired and low-cost electronic components. Figure 49 shows an image of the electronic components that are embedded in the robot body, which are: (1) three 8x8 LED arrays that are used to compose the FRED's gaze; (2) a ring with 16 RGB LEDs which are attached to the robot's chest; (3) four SG90 servomotors that are responsible for moving the FRED; (4)

⁴<https://www.tinkercad.com/things/dGqDeysEWR8-copy-of-projeto-fred-versao-2023/edit>

⁵<https://github.com/Marcelo-Rocha-UFF/fred-robot/tree/main/fred-3D-models>

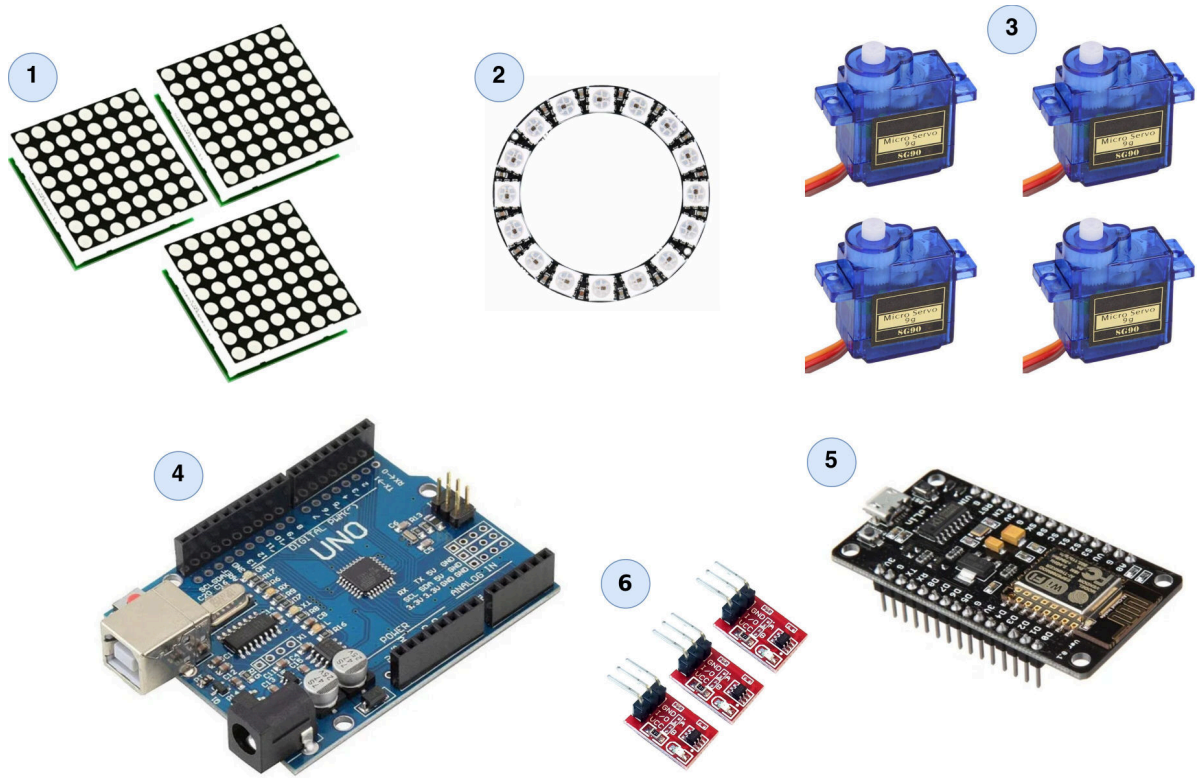


Figure 49: Hardware components that are embedded inside FRED.

an Arduino UNO board that connects directly to the elements mentioned above; (5) An ESP8266 board (NodeMCU) running LUA firmware; (6) three capacitive touch sensors.

In addition, a mini protoboard and some wires are used to connect the components to the board ports.

6.1.2.1 Electronic Schematic

Figure 50 shows the electronic schematic of the FRED hardware components assembly. As can be seen, a mini protoboard is used to assist in the connection between the power source, the components and the boards. The indication for the system power supply is a 5V DC source that can provide 2A of load. The four SG90 servomotors are connected to the Arduino digital ports that control them through PWM signals. The 8x8 LED matrices are connected in series because they are controlled by the Arduino that sends the information to the three displays in a multiplexed manner. The capacitive touch sensors work simply with on/off switches being connected to three Arduino digital ports. The RGB LEDs use a simplified connection system using only two power cables and only one data cable. As described in the FRED hardware architecture, communication between the Arduino and the ESP8266 board (NodeMCU) is done through serial communication.

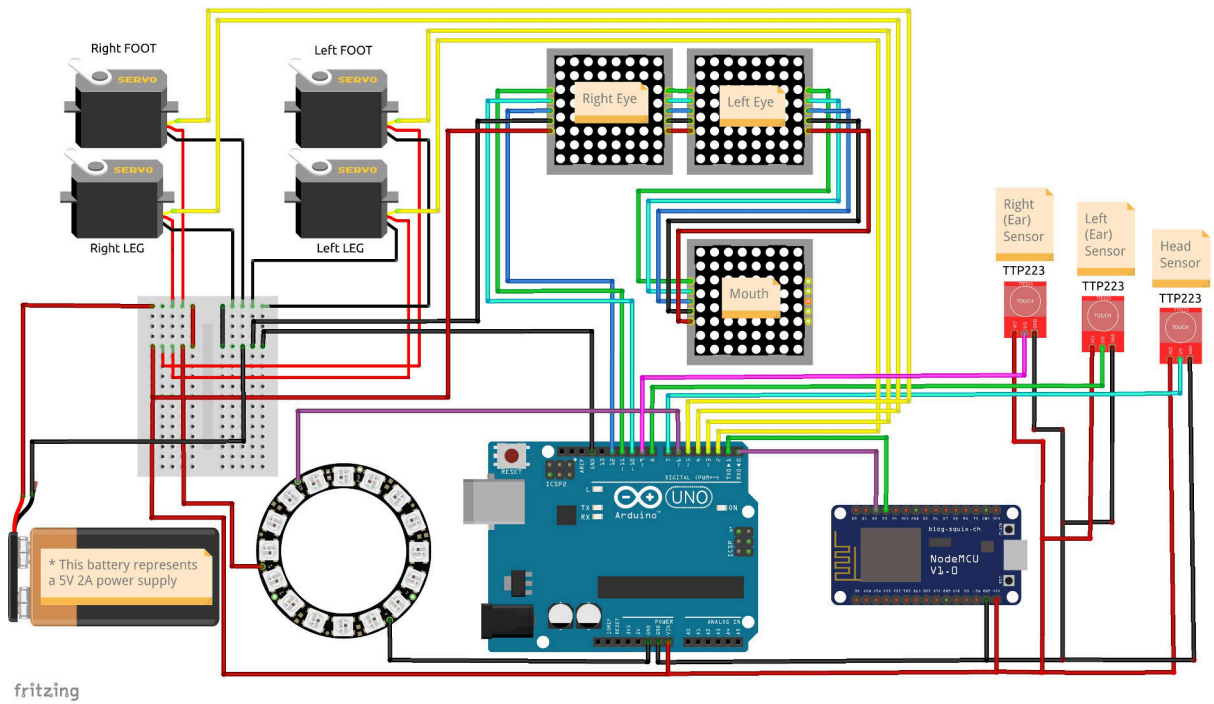


Figure 50: Electronic schematic of the assembly of the FRED components.

It is through it that the ESP8266, after receiving commands via MQTT (via wifi), decodes these commands and transforms them into bytes (sequence of characters) according to the defined communication protocol.

6.1.3 Robot Capabilities

FRED has several functionalities, including verbal and non-verbal communication elements. In addition, interaction scripts can be developed for it using the XML-based declarative programming language EvaML, described in Section A.4. These scripts run on a control software called EvaSIM, described in Section A.5. The following items present a brief description of FRED's functionalities:

- **Verbal Communication.** The FRED robot can speak using cloud resources such as IBM Watson's Text-To-Speech (TTS) service. Additionally, it can also transform speech to text using Google's Speech-To-Text (STT) API cloud service.
- **Eye Expressions.** Among the robot's nonverbal communication elements, one of the most effective is the ability to express emotion through gaze. FRED can express the following emotions through its display: anger, disgust, fear, happiness, sadness, surprise, and neutral.

- **Poses and Movements.** The robot is able to express itself through poses and movements using its legs. These elements of non-verbal communication when used in conjunction with eye expressions are able to increase FRED's expressiveness. For example, the tiptoe pose associated with the happy eye expression can convey the impression of greater happiness.
- **RGB LEDs.** According to [Bianchi and Almeida Neris \(2015\)](#), colors, in addition to being one of the most important dimensions in the process of visual comprehension, have the power to evoke emotions. FRED has a ring of RGB LEDs on its chest. In addition to using colors as an element of nonverbal communication, the robot can perform some animations with the LEDs.
- **Smart Bulb Control.** In order to make interactive sessions more attractive, especially for children, FRED has the ability to control a smart bulb using a wireless network.
- **Audio Player.** FRED can play several types of audio files such as music, sound effects and audio (speech) generated by the IBM Watson TTS service, for which it uses an external speaker.
- **Computer Vision.** Using an external webcam, the robot can identify users by recognizing their face. It can also infer the user's emotion. In addition, the robot can read QR Codes. This feature allows FRED to interact, for example, with users who have difficulty in verbal communication, allowing cards containing answers in QR code format to be used.
- **Touch Sensitivity.** As another way of interacting with the user, FRED has three capacitive touch sensors. Two of these sensors are positioned just below the robot's ears. When touched for about 5 seconds in this region, the robot reacts by smiling and bending its left or right leg, which is relative to the sensor that was touched. The third sensor is located on top of its head and, when touched for about 5 seconds, makes the robot react by bending both its legs and showing a passionate expression on its display. Since the sensors are capacitive, they are fixed internally and are not visible to users.
- **Text Emotion Recognition (TER).** The robot is able to classify the emotion contained in texts. It uses the same categorical emotion model used in the computer vision module, being able to return six emotions plus the neutral emotion.

- **Automatic Empathic Behavior (AEB).** It allows FRED to define its affective expressions, synthesizing them automatically, based on the recognition of emotions contained in a text. In order for the AEB component to fulfill its role, it makes use of the service of the TER module. The text to be classified by the TER module can be obtained in two ways: explicitly, where the programmer passes the text to be classified directly through the script, or it can be a text resulting from an STT process, as a result of interaction with the user, via voice. The AEB uses the Google Translate API. This is necessary because the text-based emotion recognition model used by the TER module was trained in English, therefore, if the input text is not in English, it must be translated into the model's language prior to classification. Once the text is translated into the appropriate language, the AEB component uses the Nonverbal Elements Control submodule to, through the emotion returned by the model, along with its probability, modulate the parameters for the robot's nonverbal communication elements (facial expression, poses and movements, and RGB LEDs), automatically synthesizing an affective expression. These parameters are passed via MQTT messages over the wireless network, reaching the hardware of the low-level layer of the FRED robot architecture.

These capabilities provide the expressive mechanisms through which this work investigates Research Question 1, concerning how a social robot can express affective states in a way that can be recognized by users.

6.1.4 Software Architecture

Since FRED uses simple hardware components with low computing power, part of its system and functionalities run on an external computer, such as a laptop. Based on this, the robot's software architecture was defined in two layers, a high-level layer and a low-level layer.

The low-level layer encompasses the software modules that run directly on the robot hardware, on the ESP8266 boards and on the Arduino UNO board. The Arduino board runs the C++ functions that handle the digital input and output ports that control the servomotors (PWM signals), the LED arrays, the touch sensors, and the RGB LED ring.

Due to the robot's limited hardware capabilities, the high-level layer of its architecture, responsible for more resource-intensive functions such as voice capture, expression recognition, etc., runs on an external computer. The next section presents this instantia-

tion.

6.2 Instantiation of FRED Control Software

This section presents another concrete use case for the proposed framework to illustrate its application. Just like the EVA control software, the FRED robotic system was also developed using the framework proposed in Chapter 3. It specifically demonstrates the instantiation of the classes available in the Robot Abstraction Layer, used to create the control software for the EVA 2.0 robot, including all its functionalities. Following EVA 2.0, GRACE, and FRED MX, it serves as a fourth real-world instantiation scenario for the proposed framework. FRED's software features modules similar to those of the EVA system, these modules belong to the same groups I, II, and III previously defined in Section 4.2.1.

As in the EVA control system, the *Script Player* module runs on a device external to the robot, such as a laptop. However, due to the restricted processing capabilities of the FRED hardware components, seven other modules run on the same external device: *TER*, *TTS*, *STT*, *Light*, *CV*, *AEB*, and *Audio*. The instantiations of these modules are not different from the same modules used in the EVA software. The mechanisms for synchronizing modules and transferring variable values are the same as those used in the EVA system, using the same special topics, *state* and *var*, respectively. The topics defined for activating the modules are basically the same as those presented in Table 4, with small variations for specific FRED functionalities. He cannot move his head or his arms, but he can move and pose using his legs. Therefore, the topics of the *Motion* module are: *pose* and *move*.

Figure 51 presents an overview of the hardware and software architecture implemented for FRED. As can be seen in figure, in the FRED system, the *broker* runs on an external device (laptop). The three FRED modules that do not run on the external device are: *Motion*, *Display* and *RGB LEDs*. They are implemented on a NodeMCU board, running a LUA firmware, allowing FRED to become an MQTT client. The NodeMCU board, in addition to allowing connection to the *broker*, allows connection and control of the Arduino board through a connection to the serial port.

The instantiation of the three modules follows the same pattern described for the modules of *Group I*, in Section 4.2.1.1. The modules of this group do not execute synchronously with other modules, that is, they do not signal the end of their tasks. Furthermore, they

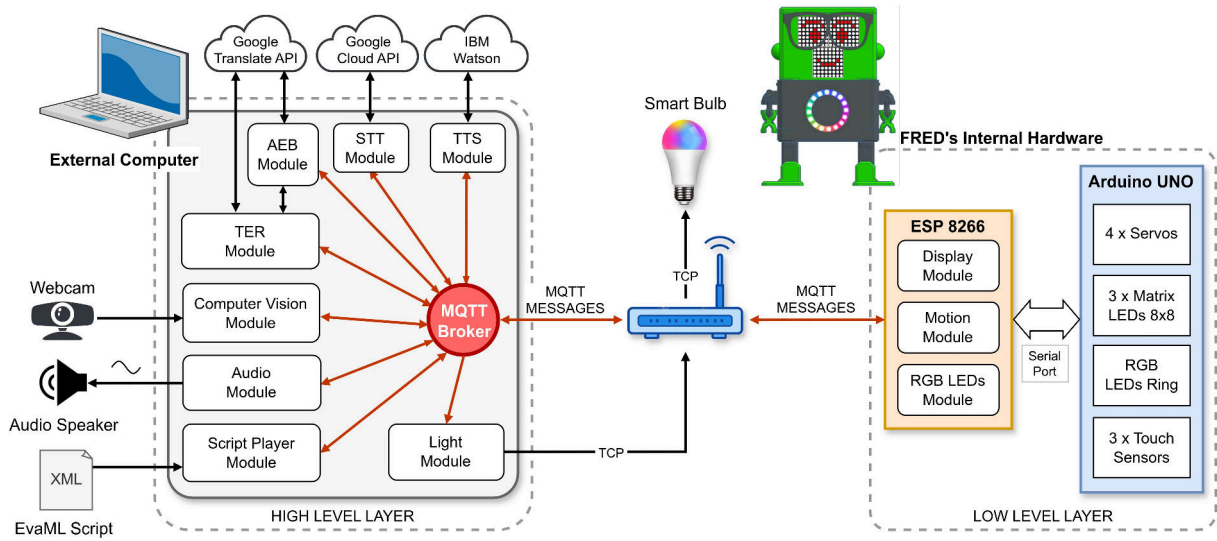


Figure 51: FRED control system modules.

do not return any type of information in text form. A video of the robots, FRED, its female counterpart Frida, and EVA 2.0, operating with the software system developed using the proposed framework is available at this [link](#).

This chapter presented a new social affective robot called FRED, a low-cost affective social robot that can become a viable option for use in healthcare and education applications in developing countries. Section 6.1.1 presented FRED with its physical characteristics and 3D models. Section 6.1.2 listed the hardware components. Section 6.1.4 presented the robot's functionalities and its software components. Finally, in Section 6.2, a new instantiation of the framework proposed in this work is made in the implementation of the FRED control system.

According [Zhou and Tian \(2020\)](#), affective expressions can significantly enhance the ability of robots to interact effectively with humans, making robots appear more socially aware and empathetic. The next chapter proposes and carries out an assessment of FRED's affective capabilities.

6.3 Evaluation of the FRED Robot and its Affective Expressions

To provide empirical evidence related to Research Question 1, this section evaluates how users perceive and recognize the affective expressions produced by FRED. This section presents the evaluation process for the FRED robot based on user perception.

Table 8: GQM goals defined for the evaluation of FRED robot.

Goal	Description
G1	Analyze FRED’s nonverbal communication capability with the purpose of evaluating its affective expressions, based on the user perception.
G2	Analyze the FRED robot with the purpose of evaluating its characteristics related to the scale Anthropomorphism, based on the user perception.
G3	Analyze the FRED robot with the purpose of evaluating its characteristics related to the scale Animacy, based on the user perception.
G4	Analyze the FRED robot with the purpose of evaluating its characteristics related to the scale Likeability, based on the user perception.
G5	Analyze the FRED robot with the purpose of evaluating its characteristics related to the scale Perceived Intelligence, based on the user perception.
G6	Analyze the FRED robot with the purpose of evaluating whether, based on the user’s perception, children would like to interact with FRED robot.

The evaluation of the FRED robot focuses on its ability to convey affective expressions and how those expressions are perceived by users. This includes analyzing the participants’ ability to recognize the emotions expressed by the robot, as well as the perception of its characteristics based on the Godspeed Questionnaire Series (GQS) dimensions: Anthropomorphism, Animacy, Likeability, and Perceived Intelligence. Supplementary qualitative comments are also analyzed to gain insights into user experiences. Section 6.3.1 presents the methodology used to develop and structure this evaluation phase. Section 6.3.2 presents the details of the experiment. Section 6.3.3 presents the results obtained, and Section 6.3.4 analyzes and discusses them.

6.3.1 Methodology

The evaluation of the FRED robot was conducted using the Goal Question Metric (GQM) (CALDIERA; ROMBACH, 1994) paradigm, combined with the Godspeed Questionnaire Series (GQS) (BARTNECK et al., 2009). This approach provides a structured framework to define evaluation goals, derive research questions, and establish corresponding metrics. Six goals (G1 to G6) were defined to assess different aspects of the robot from the users’ perspective, including its affective expression capability, perceived characteristics, and expected acceptance by children. Table 8 summarizes these goals. To achieve these goals, two types of instruments were employed.

For G1, a set of questions was developed to assess participants’ perception of four basic emotions, happiness, sadness, anger, and surprise, as presented in Table 9. For

Table 9: Questions for Goal G1.

Question	Description
Q1	Did you perceive when the robot expressed happiness?
Q2	Did you perceive when the robot expressed sadness?
Q3	Did you perceive when the robot expressed anger?
Q4	Did you perceive when the robot expressed surprise?

Table 10: Questions for Goal G2.

Question	Description
Q5, Q6, Q7, Q8, Q9	Rate your impression of the robot's human characteristics on the Anthropomorphism scale according to GQS.

G2 to G5, standardized questions from the GQS were used to assess the dimensions of Anthropomorphism, Animacy, Likeability, and Perceived Intelligence, as presented in Tables 10, 11, 12 and 13. For G6, as shown in Table 14, a single question was defined to evaluate perceived child acceptance.

Each Goal is associated with a specific metric. For G1, the Percentage of Expected Answers (PEA) is shown in Table 15. It is computed as the ratio between the number of responses that match the intended emotion expressed by the robot and the total number of response provided by participants for that emotion. Since affective perception is inherently subjective, the notion of "expected answer" is used instead of "correct answer", reflecting the intended emotional expression designed for each video segment. The PEA metric is formally defined as follows. Let $R_e = \{r_1, r_2, \dots, r_n\}$ be the set of responses for emotion e . Then:

$$PEA(e) = \frac{|\{r_i \in R_e \mid r_i = e\}|}{|R_e|} \quad (6.1)$$

For G2 to G5, the corresponding GQS dimensions were used as quantitative metrics. Tables 16, 17, 18 and 19 present, respectively, the descriptions of the metrics. For G6, the Perceived Child Acceptance (PCA) metric was defined and is described in Table 20.

Table 11: Questions for Goal G3.

Question	Description
Q10, Q11, Q12, Q13, Q14, Q15	Rate your impression of the robot's human characteristics on the Animacy scale according to GQS.

Table 12: Questions for Goal G4.

Question	Description
Q16, Q17, Q18, Q19, Q20	Rate your impression of the robot's human characteristics on the Likeability scale according to GQS.

Table 13: Questions for Goal G5.

Question	Description
Q21, Q22, Q23, Q24, Q25	Rate your impression of the robot's human characteristics on the Perceived Intelligence scale according to GQS.

Table 14: Questions for Goal G6.

Question	Description
Q26	Children would like to interact with the FRED robot.

Table 15: Metrics for G1 questions.

Metric	Description	Question
PEA	<i>Percentage of Expected Answers</i> is computed as the ratio between the number of responses that match the intended emotion expressed by the robot and the total number of responses provided by participants for that expression. The user responds using a 3-point Likert scale, where 0 means "No Emotion", 1 means "Mild Emotion" and the 2 means "Intense Emotion".	Q1, Q2, Q3, Q4.

Table 16: Metrics for G2 questions.

Metric	Description	Question
AT	Anthropomorphism refers to the attribution of a human form, human characteristics, or human behavior to nonhuman things such as robots, computers, and animals (Bartneck et al, 2023). It will be measured through five anthropomorphic characteristics using the five-point Likert scale.	Q5, Q6, Q7, Q8, Q9

Table 17: Metrics for G3 questions.

Metric	Description	Question
AN	Animacy is defined as the perception that a robot has characteristics that can be attributed to living beings (Bartneck et al, 2023). This includes physical behaviors, reactions to stimuli, and language skills, which are typically associated with animals. It will be measured through six characteristics related to animacy using the five-point Likert scale.	Q10, Q11, Q12, Q13, Q14, Q15

Table 18: Metrics for G4 questions.

Metric	Description	Question
LK	Likeability can be understood in the context of human-robot interaction as the perception that a robot is friendly, pleasant or worthy of empathy from humans (Bartneck et al, 2023). It will be measured through five characteristics related to likeability using the five-point Likert scale.	Q16, Q17, Q18, Q19, Q20

Table 19: Metrics for G5 questions.

Metric	Description	Question
PI	Perceived Intelligence is defined as the ability of a robot to act intelligently and is evaluated based on how humans perceive the robot's behavior in relation to characteristics such as competence, knowledge and responsibility (Bartneck et al, 2023). It will be measured through five characteristics related to perceived intelligence using the five-point Likert scale.	Q21, Q22, Q23, Q24, Q25

Table 20: Metrics for G6 questions

Metric	Description	Question
PCA	<i>Perceived Child Acceptance</i> measures users' perception of children's interest in interacting with the FRED robot. It will be measured using user answer with a five-point Likert scale, where 1 means "Strongly Disagree" and the 5 means "Strongly Agree".	Q26

6.3.2 Experiment

An experiment was conducted to evaluate the FRED robot from the users' perspective, focusing on its affective expression capability, perceived characteristics, and expected acceptance by children. For this purpose, a video lasting 2 minutes and 48 seconds was recorded, presenting a dialogue between the robots FRED and its customized version, Frida. The video was segmented into six blocks with varying durations, in which the robot expressed different affective states through nonverbal communication elements, including facial expressions, body poses, and RGB LED patterns. Participants were asked to watch these video blocks and report their perceptions through an online questionnaire.

6.3.2.1 Participants and Recruitment

Participants were recruited from diverse professional backgrounds through the dissemination of a link containing the experimental materials and questionnaire. The sample comprised 113 volunteers, including 71 males (62.8%) and 42 females (37.2%). Regarding age distribution, most participants were between 25 and 65 years old, with smaller groups under 25 and over 65 years of age. In terms of professional background, 51 (45.1%) participants reported working in technology-related fields (e.g., computer science and software engineering), while the 62 remaining participants (54.9%) were from non-technical areas, including healthcare, education, and the humanities. The majority of participants were from Brazil ($n = 109$), with additional participants from Mexico, Spain, and Switzerland.

This study was approved by the Ethical Research Committee of Plataforma Brasil, maintained by the Brazilian Ministry of Health (CAAE 92434225.1.0000.5243).

6.3.2.2 Instruments and Data Collection

The experiment was carried out remotely through a web browser, where each participant watched pre-recorded videos and then answered a questionnaire with questions related to Goal G1, another questionnaire related to Goals G2 to G5 and a specific question related to Goal G6. These questionnaires were created using the *Google Forms*⁶ application. At the end of the questionnaires, participants could leave comments and suggestions regarding their impressions of FRED and its affective expressions. These responses were later examined to provide complementary qualitative insights into participants' perceptions.

⁶<https://www.google.com/forms/about>

6.3.2.3 Experiment Procedure

To evaluate the affective expressions of the FRED robot, participants were presented with a video depicting a dialogue between the FRED robot and its customized version, Frida. In this interaction, both robots expressed emotions through nonverbal communication elements, including facial expressions displayed on the screen, body poses, and RGB LED patterns. The complete video was segmented into six blocks, each containing one or more affective expressions performed by the robot. This segmentation was adopted to facilitate the evaluation process, allowing participants to focus on shorter interaction excerpts. The full video and the corresponding segmented blocks, used as stimuli in the experiment, are publicly available in a dedicated online repository and can be accessed via this [link](#)⁷.

The dialogue was initially generated using a large language model based on a prompt designed to elicit sentences with different emotional tones. The generated content was manually refined to ensure the inclusion of the target emotions, happiness, sadness, anger, and surprise, as well as to improve linguistic coherence.

To generate the robot's affective behavior, the textual content of each utterance was processed by a Text Emotion Recognition (TER) module, which identified the dominant emotion and its associated probability. These outputs were then used by the Automatic Empathic Behavior (AEB) component to modulate the robot's nonverbal expressions. Specifically, the robot's facial display, LED colors, and body posture were adjusted according to the inferred emotion and its intensity level. Each emotion was represented with up to two intensity levels, corresponding to "Mild Emotion" and "Intense Emotion", depending on the probability returned by the TER module. The mapping between emotion classes, probability values, and nonverbal expressions is presented in Table 21. Examples of the resulting affective expressions are illustrated in Figure 52.

The AEB component used in this experiment corresponds to an earlier version instantiated for FRED, presented in Section 6.2. This version employs a predefined mapping to directly control FRED's non-verbal communication elements. It is coupled to the TER module and does not use the robot's affective state as a parameter in the mapping, relying instead only on the emotion identified in the text of its speech. Furthermore, it directly controls the execution of the behavior on the robot. These characteristics distinguish it from the extensible AEB component proposed as part of the Affective Layer introduced in this thesis in Chapter 3.

⁷<https://github.com/marcelomrocha/fred-robot/tree/main/fred-exp-eval-links>

Table 21: Mapping of nonverbal communication elements used by FRED for each emotion, based on the probability value of the emotion classified in the Text Emotion Recognition (TER) module in the text.

Emotion	Probability Range	Intensity Level	Nonverbal Communication Elements		
			Display	RGB Leds	Pose
Happiness	0% $\rightarrow$ 95%	Mild Happiness	Happy	Green	Neutral
	95% $\rightarrow$ 100%	Intense Happiness	Happy	Green	Tiptoe
Sadness	0% $\rightarrow$ 95%	Mild Sadness	Sad	Blue	Broken left foot
	95% $\rightarrow$ 100%	Intense Sadness	Sad	Blue	Broken feet
Anger	0% $\rightarrow$ 95%	Mild Anger	Angry	Red	Neutral
	95% $\rightarrow$ 100%	Intense Anger	Angry2	Red	Neutral
Surprise	0% $\rightarrow$ 95%	Mild Surprise	Surprise	Yellow	Neutral
	95% $\rightarrow$ 100%	Intense Surprise	Surprise	Yellow	Broken feet
Neutral	0% $\rightarrow$ 100%	Neutral	Neutral	Off	Neutral

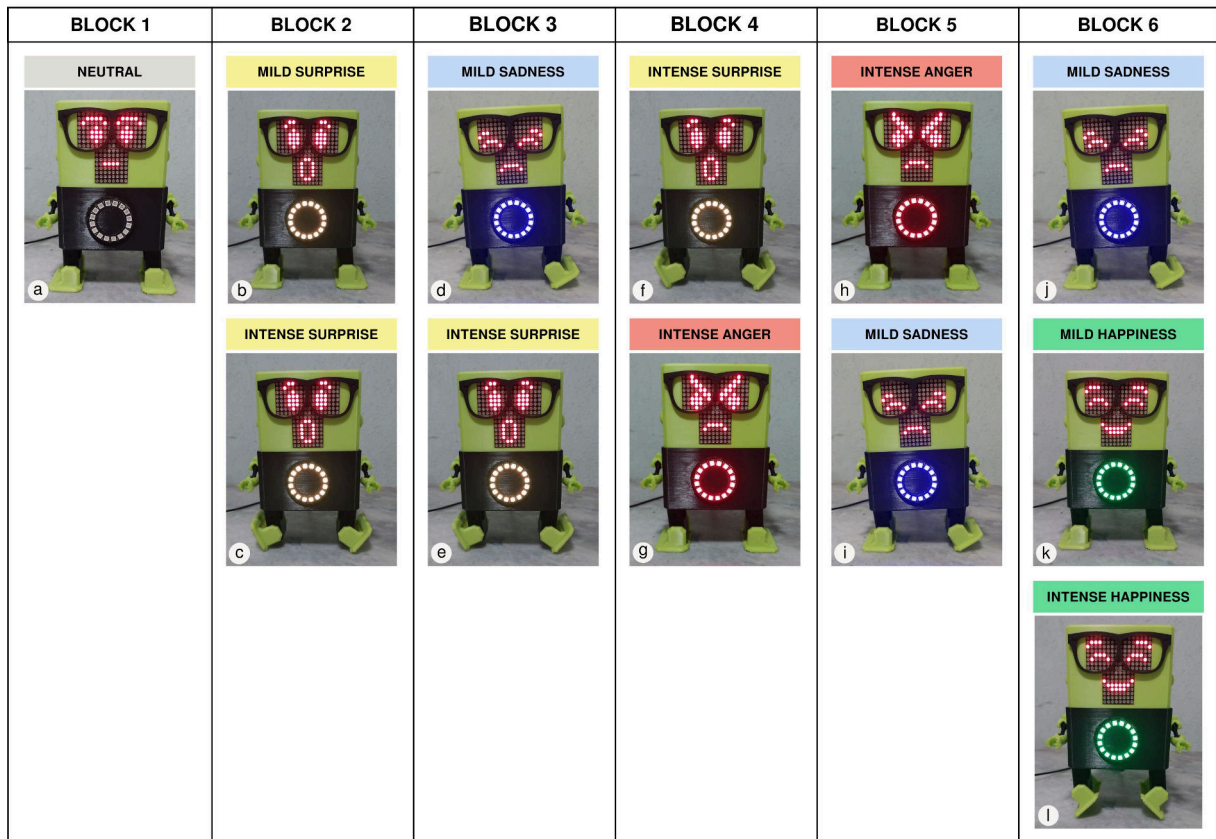


Figure 52: Affective expressions of the FRED robot presented to participants across the six video blocks. Each expression is generated through the combination of nonverbal communication elements, facial display, body pose, and RGB LED patterns, modulated according to the emotion class and its associated probability inferred from the text emotion recognition model. The neutral expression (a) represents the absence of emotion.

Participants were first instructed to watch the complete video to understand the context of the interaction. Subsequently, they evaluated each of the six video blocks individually. For each block, participants were asked to indicate their perception of the intensity of four emotions, happiness, sadness, anger, and surprise, expressed by the FRED robot. For each emotion, participants selected one of three possible responses: "No Emotion", "Mild Emotion", or "Intense Emotion". If an emotion was not perceived in a given block, the expected response was "No Emotion". In total, each participant answered 24 questions, corresponding to four emotions evaluated across six video blocks.

The complete dialogue between the robots, including all utterances, their corresponding video blocks, the inferred emotions, and the associated probability values, is presented in Table 22. This table provides a comprehensive reference for the construction of the stimuli and supports the interpretation of the experimental results.

Table 22: Sentences from the complete dialogue between robots FRED and Frida with the indication of the video blocks in which they were spoken and their respective classifications returned by the TER module containing the emotion, with the highest probability and its value.

#	Robot	Sentence	Video Block	Inferred Emotion	Probability
1	Frida	Hello FRED, how are you?	B1	Neutral	95.2%
2	FRED	Hi Frida, I'm fine and you?	B1	Neutral	96.2%
3	Frida	I'm fine too, thanks. But I have news.	B1, B2	Neutral	80.2%
4	FRED	What happened?	B2	Surprise	73.3%
5	Frida	I lost my phone yesterday.	B2	Sad	84%
6	FRED	Seriously? So sorry, Frida. Where did it happen?	B2, B3	Surprise	96.2%
7	Frida	I think it was on the bus. I am so sad.	B3	Sad	97.8%
8	FRED	I can imagine. Losing a phone is terrible.	B3	Sad	93.2%
9	Frida	Yes, there were many photos and important messages.	B3	Neutral	97.2%
10	FRED	Wow, what a shame. Have you tried calling it?	B3, B4	Surprise	96.5%
11	Frida	Yes, but no one answered. I'm starting to get irritated.	B4	Anger	42.5%
12	FRED	Yeah... I would also be very angry to lose something so important.	B4	Anger	97.5%
13	Frida	And the sad thing is that it was a birthday present.	B5	Sad	73.7%
14	FRED	Wow, that's even worse. Very frustrating.	B5	Sad	73.6%
15	Frida	But you know, I decided I'm going to buy another one today.	B6	Neutral	65%
16	FRED	Great, Frida! Buying something new always cheers us up.	B6	Happy	98.7%
17	Frida	True, I'm actually excited to choose a new model.	B6	Happy	93.6%
18	FRED	That's right! It's great to get a new phone.	B6	Happy	98%
19	Frida	Thanks for listening, FRED. You always help me.	B6	Neutral	74.9%
20	FRED	Of course, Frida! Good luck with your purchase!	B6	Happy	69.6%
21	Frida	Thanks, FRED! We'll talk later. Bye bye!	B6	Neutral	52.2%

6.3.2.4 Data Analysis

A quantitative analysis was conducted on the data collected from participants. For G1, the Percentage of Expected Answers (PEA) was used to assess the alignment between participants' responses and the intended emotion expressed by the robot.

For G2 to G5, corresponding to the dimensions of the Godspeed Questionnaire Series (GQS), participants' responses were aggregated across items to obtain representative scores for each dimension, Anthropomorphism (AT), Animacy (AN), Likeability (LK), and Perceived Intelligence (PI).

For G6, the Perceived Children Acceptance (PCA) metric was derived from participants' Likert-scale responses to the corresponding question.

The data were further analyzed across participant groups defined by gender (male and female) and professional background (technological and non-technological). Prior research in Human-Robot Interaction (HRI), including the studies by [Bretschneider et al. \(2022\)](#) and [Chita-Tegmark, Lohani, and Scheutz \(2019\)](#) and the works of [Jirak et al. \(2022\)](#) and [Rosén et al. \(2024\)](#), supports the assumption that user-related factors, such as gender and familiarity with technology, may influence how robot behavior and affective cues are perceived and interpreted. To investigate potential differences between these groups, a non-parametric Mann–Whitney test was applied, considering comparisons between gender groups and professional background groups. Furthermore, the qualitative feedback obtained through open-ended responses was analyzed to identify recurring patterns in the participants' perceptions.

6.3.2.5 Reliability Analysis of the Godspeed Questionnaire Series (GQS)

The internal consistency of the GQS was assessed using Cronbach's alpha coefficient. The coefficient was computed for each dimension individually, as well as for all questionnaire items combined. The interpretation of the values followed the classification proposed in ([GEORGE; MALLERY, 2019](#)): (≥ 0.9 : Excellent), (≥ 0.8 and < 0.9 : Good), (≥ 0.7 and < 0.8 : Acceptable), (≥ 0.6 and < 0.7 : Questionable), (≥ 0.5 and < 0.6 : Bad) and (< 0.5 : Unacceptable). Table 23 presents the results of this evaluation.

Table 23: Cronbach's α values for each dimension of the GQS and for all items combined.

Godspeed series	Cronbach's α	Classification
Anthropomorphism	0.83	Good internal consistency
Animacy	0.86	Good internal consistency
Likeability	0.95	Excellent internal consistency
Perceived Intelligence	0.94	Excellent internal consistency
All Items	0.94	Excellent internal consistency

6.3.3 Results

This section presents the results obtained from the evaluation of the FRED robot, based on the experimental study and the questionnaires designed to address the defined research goals. The analysis includes the assessment of the robot's affective expression capability using the PEA metric, as well as the evaluation of perceived characteristics using the GQS, and the assessment of perceived child acceptance using the PCA metric. Additionally, qualitative feedback provided by participants is analyzed to complement the quantitative findings. The non-parametric Mann–Whitney test was used to compare groups and assess differences in participants' perceptions.

The following sections present the results for each defined goal (G1 to G6).

6.3.3.1 G1 – Affective Expression Capability of FRED

To address G1, the PEA metric was calculated for each question and used to evaluate how participants' responses matched the intended emotion expressed by the robot across the six video blocks. Finally, participants were divided into groups based on gender (Male: 71 and Female: 42) and professional background (Tech: 52 and Non-Tech: 61). From these data, the non-parametric Mann-Whitney test was used to determine statistically significant differences between the responses of the groups.

The question related to the perception of happiness was used to evaluate FRED's ability to express this emotion. Participants' responses showed a high level of agreement with the expected answers across most video blocks, with values above 87% for blocks 2 to 6. In these blocks, participants identified both the absence of happiness and its most intense expression. In Block 1, 58.4% of participants selected the expected response "No Happiness".

The results for sadness indicate lower agreement with the expected responses when compared to the other emotions, ($PEA = 49.71\%$). In Block 1, 76.1% of participants

Table 24: Distribution of participants' responses for questions Q1 to Q4 across all video blocks, including expected answers and PEA values, based on all participants ($n = 113$).

EMOTION EVALUATED	VIDEO BLOCK	EXPECTED ANSWER	ALL USER'S ANSWERS			PERCENTAGE OF EXPECTED ANSWERS
			NO	MILD	INTENSE	
HAPPINESS (Q1)	B1	NO HAPPINESS	58.4%	31.9%	9.7%	84.81%
	B2	NO HAPPINESS	88.5%	7.1%	4.4%	
	B3	NO HAPPINESS	91.2%	4.4%	4.4%	
	B4	NO HAPPINESS	92%	3.5%	4.4%	
	B5	NO HAPPINESS	91.2%	6.2%	2.7%	
	B6	INTENSE HAPPINESS	4.4%	8%	87.6%	
SADNESS (Q2)	B1	NO SADNESS	76.1%	17.7%	6.2%	49.71%
	B2	NO SADNESS	56.6%	34.5%	8.8%	
	B3	MILD SADNESS	5.3%	39.8%	54.9%	
	B4	NO SADNESS	61.9%	29.2%	8.8%	
	B5	MILD SADNESS	7.1%	37.2%	55.8%	
	B6	MILD SADNESS	61.1%	26.5%	12.4%	
ANGER (Q3)	B1	NO ANGER	83.2%	7.1%	9.7%	76.4%
	B2	NO ANGER	81.4%	11.5%	7.1%	
	B3	NO ANGER	74.3%	12.4%	13.3%	
	B4	INTENSE ANGER	1.8%	11.5%	86.7%	
	B5	INTENSE ANGER	30.1%	28.3%	41.6%	
	B6	NO ANGER	91.2%	5.3%	3.5%	
SURPRISE (Q4)	B1	NO SURPRISE	69.9%	23%	7.1%	62.68%
	B2	INTENSE SURPRISE	1.8%	18.6%	79.6%	
	B3	INTENSE SURPRISE	7.1%	30.1%	62.8%	
	B4	INTENSE SURPRISE	46.9%	34.5%	18.6%	
	B5	NO SURPRISE	77.9%	11.5%	10.6%	
	B6	NO SURPRISE	67.3%	22.1%	10.6%	

identified the absence of sadness, "No Sadness". In Block 2, the proportion of expected responses decreased to 56.6%. In Blocks 3 and 5, where mild sadness was expected, participants identified the emotion as "Intense Sadness", corresponding to 54.9% and 55.8% of responses, respectively. In Block 4, 61.9% of participants correctly identified the absence of sadness. However, in Block 6, where mild sadness was expected, the of participants (61.1%) reported the absence of sadness.

The PEA value for anger was 76.4%. In Blocks 1, 2, 3, and 6, where the absence of anger was expected, 83.2%, 81.4%, 74.3%, and 91.2% of participants, respectively, identified the absence of this emotion, selecting "No Anger". In Blocks 4 and 5, where "Intense Anger" was expected, 86.7% of participants selected the expected answer. In Block 5, 41.6% of responses indicated "Intense Anger", while the remaining responses were distributed between "No Anger" and "Mild Anger".

For surprise, the PEA value was 62.68%. In Blocks 1, 5, and 6, where the absence of surprise was expected, 69.9%, 77.9%, and 67.3% of participants, respectively, identified this condition, selecting "No Surprise". In Blocks 2 and 3, where intense surprise was expected, the participants identified the highest intensity of the emotion, with 79.6% in Block 2 and 62.8% in Block 3, selecting "Intense Surprise". In Block 4, although a intense surprise was expected, 46.9% of participants identified the absence of surprise, selecting "No surprise".

Group Comparison by Gender. Table 25 presents the median values, interquartile ranges, and results of the Mann-Whitney tests used in the comparisons between the four groups for the PEA for each Block. Regarding gender, no statistically significant differences were observed between the groups ($p > 0.05$). Furthermore, the effect sizes were small ($r \leq 0.15$).

Group Comparison by Professional Background. Regarding professional training, statistically significant differences ($p < 0.05$) were observed in the seven blocks described below.

For the emotion sadness, in Block 1, both groups presented identical median and interquartile range values ($M_d = 0$, $IQR = 1$). The statistical test resulted in $U = 1321.5$, $p = 0.041$, and $r = 0.17$.

For anger, statistically significant differences were identified in Blocks 1, 2, and 3. For Block 1, both groups showed the same median and interquartile range ($M_d = 0$, $IQR = 0$)

and the statistical test results were $U = 1315$, $p = 0.017$, and $r = 0.17$. In Blocks 2 and 3, both groups showed the same median and interquartile range ($M_d = 0$, $IQR = 1$) and the statistical test results were, respectively, $U = 1315$, $p = 0.017$, and $r = 0.17$, and $U = 1316.5$, $p = 0.043$, and $r = 0.17$.

For surprise, statistically significant differences were observed in Blocks 1, 4, and 6. In Block 1, both groups presented the same median ($M_d = 0$), whereas the interquartile range differed, with $IQR = 0$ for the Tech group and $IQR = 1$ for the Non-Tech group. The statistical test results for the block were $U = 1295$, $p = 0.037$, and $r = 0.18$. In Block 4, the median and interquartile range values for the Tech and Non-Tech groups were, respectively, ($M_d = 1$, $IQR = 2$) and ($M_d = 0$, $IQR = 1$) and the statistical test results for the block were, respectively, $U = 1254$, $p = 0.038$, and $r = 0.20$. In Block 6, both groups presented identical median values ($M_d = 0$), while the interquartile ranges differed, with $IQR = 0$ for the Tech group and $IQR = 1$ for the Non-Tech group. For the group, the Mann–Whitney test resulted in $U = 1115.5$, $p = 0.001$, and $r = 0.30$.

These results provide evidence related to Research Question 1. The findings show that the affective states expressed by FRED through its non-verbal communication elements can be recognized by users, although the level of recognition varies across the affective states evaluated. The qualitative findings complement these results by indicating that the clarity of affective expressions may also depend on specific expressive elements, particularly facial expressions, body movements, and their coordination.

Table 25: Median values calculated for the answers of all participants, for males and females, for participants in the technological and non-technological areas, as well as the p -values resulting from the non-parametric Mann-Whitney test when comparing the answers of the groups by gender and by professional background.

EMOTION BEING EVALUATED	VIDEO BLOCK	ALL (113) PARTICIPANTS		BY GENDER						BY PROFESSIONAL BACKGROUND					
				MALE (71)		FEMALE (42)		Mann-Whitney		TECH (52)		NON-TECH (61)		Mann-Whitney	
		M_d	IQR	M_d	IQR	M_d	IQR	U	p	M_d	IQR	M_d	IQR	U	p
HAPPINESS	B1	0	1	0	1	0	1	1470	0.889	0	1	1	1	1297.5	0.058
	B2	0	0	0	0	0	0	1373	0.207	0	0	0	0	1416.5	0.079
	B3	0	0	0	0	0	0	1475	0.852	0	0	0	0	1493	0.279
	B4	0	0	0	0	0	0	1450.5	0.613	0	0	0	0	1460	0.123
	B5	0	0	0	0	0	0	1426	0.436	0	0	0	0	1441	0.091
	B6	2	0	2	0	2	0	1427	0.509	2	0	2	0	1553	0.743
SADNESS	B1	0	0	0	1	0	1	1448	0.734	0	1	0	1	1321.5	0.041*
	B2	0	1	0	0	0	0	1283.5	0.163	0	1	0	1	1388.5	0.198
	B3	2	1	2	0	1.5	0	1362	0.385	2	1	2	1	1488	0.523
	B4	0	1	0	0	0	0	1352	0.338	0	1	0	1	1386.5	0.182
	B5	2	1	2	0	2	0	1391	0.502	2	1	2	1	1551	0.821
	B6	0	1	0	0	0	0	1392	0.500	0	1	0	1	1366	0.145
ANGER	B1	0	0	0	0	0	0	1434.5	0.609	0	0	0	0	1315	0.017*
	B2	0	0	0	0	0	0	1438.5	0.648	0	0	0	1	1313	0.020*
	B3	0	1	0	1	0	0	1444.5	0.721	0	0	0	1	1316.5	0.043*
	B4	2	0	2	0	2	0	1413	0.434	2	0	2	0	1532	0.600
	B5	1	2	1	1.5	1	2	1322	0.285	1	1.25	1	2	1444.5	0.386
	B6	0	0	0	0	0	0	1468	0.786	0	0	0	0	1435	0.078
SURPRISE	B1	0	1	0	1	0	1	1419	0.597	0	0	0	1	1295	0.037*
	B2	2	0	2	0	2	0	1396.5	0.424	2	0	2	1	1388	0.104
	B3	2	1	2	1	2	1	1470	0.886	2	1	2	1	1502	0.572
	B4	1	1	0	1	1	1	1391	0.521	1	2	0	1	1254	0.038*
	B5	0	0	0	0	0	0	1470.5	0.870	0	0	0	1	1377.5	0.098
	B6	0	1	0	1	0	0	1274.5	0.121	0	0	0	1	1115.5	0.001*

6.3.3.2 G2 – Perceived Anthropomorphism

To address G2, which aims to evaluate the FRED robot with respect to the Anthropomorphism dimension based on user perception, the AT metric was computed from participants' responses to questions Q5 to Q9. Table 26 presents the detailed results for all GQS dimensions (G2 to G5), including the medians and interquartile ranges for each semantic item. The resulting overall values of AT metric were $M_d = 3$ and $IQR = 1$.

Group Comparison by Gender. For all GQS dimensions (G2 to G5), detailed group comparison results are presented in Tables 27 and 28, corresponding to comparisons by gender and professional background, respectively. For the Anthropomorphism dimension, both groups presented the same median value ($M_d = 3$), with interquartile ranges of $IQR = 0$ for the Male group and $IQR = 1$ for the Female group. The statistical test results indicated no statistically significant difference between the groups ($U = 12$, $p = 1.00$, $r = 0.04$).

Group Comparison by Professional Background. The Tech group showed a median of $M_d = 2$ and $IQR = 1$, whereas the Non-Tech group showed $M_d = 3$ and $IQR = 0$. The statistical test indicated no statistically significant difference between the groups ($U = 4$, $p = 0.056$, $r = 0.680$).

6.3.3.3 G3 – Perceived Animacy

To address G3, which focuses on evaluating the FRED robot with respect to the Animacy dimension based on user perception, the AN metric was computed from participants' responses to questions Q10 to Q15. The resulting overall values of AN metric were $M_d = 3.5$ and $IQR = 1.75$.

Group Comparison by Gender. For the Animacy dimension, the Male group presented a median of $M_d = 3$ and $IQR = 1.5$, whereas the Female group showed $M_d = 3.5$ and $IQR = 1$. The statistical test resulted in $U = 14$, $p = 0.55$ and $r = 0.22$, indicating no statistically significant difference between the groups.

Group Comparison by Professional Background. The Tech group showed $M_d = 3$ and $IQR = 1.5$, while the Non-Tech group presented $M_d = 3.5$ and $IQR = 1$. The statistical

Table 26: Values of the metrics (AT, AN, LK and PI) for all participants and the statistical values resulting from the Mann-Whitney test.

ALL PARTICIPANTS (113)				
GOAL G2 (QUESTION)	SERIES: ANTHROPOMORPHISM			
	SEMANTIC ITEM	M_d	IQR	METRIC (AT)
Q5	Fake – Natural	3	2	3
Q6	Machinelike – Humanlike	2	2	
Q7	Unconscious – Conscious	4	2	
Q8	Artificial – Lifelike	3	2	
Q9	Moving rigidly – Moving elegantly	2	2	
GOAL G3 (QUESTION)	SERIES: ANIMACY			
	SEMANTIC ITEM	M_d	IQR	METRIC (AN)
Q10	Dead – Alive	4	2	3.5
Q11	Stagnant – Lively	3	1	
Q12	Mechanical – Organic	2	2	
Q13	Artificial – Lifelike	2	2	
Q14	Inert – Interactive	3	1	
Q15	Apathetic – Responsive	3	2	
GOAL G4 (QUESTION)	SERIES: LIKEABILITY			
	SEMANTIC ITEM	M_d	IQR	METRIC (LK)
Q16	Dislike – Like	4	2	5
Q17	Unfriendly – Friendly	5	1	
Q18	Unkind – Kind	5	1	
Q19	Unpleasant – Pleasant	5	1	
Q20	Awful – Nice	5	1	
GOAL G5 (QUESTION)	SERIES: PERCEIVED INTELLIGENCE			
	SEMANTIC ITEM	M_d	IQR	METRIC (PI)
Q21	Incompetent – Competent	4	2	4
Q22	Ignorant – Knowledgeable	4	2	
Q23	Irresponsible – Responsible	4	1	
Q24	Unintelligent – Intelligent	4	2	
Q25	Foolish – Sensible	4	1	

Table 27: Values of the metrics (AT, AN, LK and PI) for the Male and Female groups and the statistical values resulting from the Mann-Whitney test.

SERIES: ANTHROPOMORPHISM									
GOAL G2 (QUESTION)	SEMANTIC ITEM	BY GENDER							
		MALE (71)			FEMALE (42)			Mann-Whitney	
		M_d	IQR	METRIC (AT)	M_d	IQR	METRIC (AT)	U	p
Q5	Fake – Natural	3	2	3	3	1	3	12	1
Q6	Machinelike – Humanlike	3	1		2	2			
Q7	Unconscious – Conscious	3	1		4	2			
Q8	Artificial – Lifelike	3	1.5		3	2			
Q9	Moving rigidly – Moving elegantly	2	2		2	2			
SERIES: ANIMACY									
GOAL G3 (QUESTION)	SEMANTIC ITEM	BY GENDER							
		MALE (71)			FEMALE (42)			Mann-Whitney	
		M_d	IQR	METRIC (AN)	M_d	IQR	METRIC (AN)	U	p
Q10	Dead – Alive	4	2	3	4	1.75	3.5	14	0.55
Q11	Stagnant – Lively	3	1		3	1			
Q12	Mechanical – Organic	2	2		2	1			
Q13	Artificial – Lifelike	2	1		3	2			
Q14	Inert – Interactive	3	2		4	2			
Q15	Apathetic – Responsive	4	2		4	1			
SERIES: LIKEABILITY									
GOAL G4 (QUESTION)	SEMANTIC ITEM	BY GENDER							
		MALE (71)			FEMALE (42)			Mann-Whitney	
		M_d	IQR	METRIC (LK)	M_d	IQR	METRIC (LK)	U	p
Q16	Dislike – Like	4	2	4	5	1	5	0	0.003*
Q17	Unfriendly – Friendly	4	1		5	1			
Q18	Unkind – Kind	4	1		5	0.75			
Q19	Unpleasant – Pleasant	4	1		5	0			
Q20	Awful – Nice	4	1		5	1			
SERIES: PERCEIVED INTELLIGENCE									
GOAL G5 (QUESTION)	SEMANTIC ITEM	BY GENDER							
		MALE (71)			FEMALE (42)			Mann-Whitney	
		M_d	IQR	METRIC (PI)	M_d	IQR	METRIC (PI)	U	p
Q21	Incompetent – Competent	4	2	4	4	1	4	10	0.424
Q22	Ignorant – Knowledgeable	4	2		4	1			
Q23	Irresponsible – Responsible	4	2		4	1			
Q24	Unintelligent – Intelligent	4	2		4	1.75			
Q25	Foolish – Sensible	4	2		5	1			

Table 28: Values of the metrics (AT, AN, LK and PI) for the Tech and Non-Tech groups and the statistical values resulting from the Mann-Whitney test.

SERIES: ANTHROPOMORPHISM									
GOAL G2 (QUESTION)	SEMANTIC ITEM	BY PROFESSIONAL BACKGROUND							
		TECH (52)			NON-TECH (61)			Mann-Whitney	
		M_d	IQR	METRIC (AT)	M_d	IQR	METRIC (AT)	U	p
Q5	Fake – Natural	3	2	2	3	1	3	4	0.056
Q6	Machinelike – Humanlike	2	2		3	2			
Q7	Unconscious – Conscious	3	1		4	2			
Q8	Artificial – Lifelike	2	2		3	2			
Q9	Moving rigidly – Moving elegantly	2	1.25		3	2			
SERIES: ANIMACY									
GOAL G3 (QUESTION)	SEMANTIC ITEM	BY PROFESSIONAL BACKGROUND							
		TECH (52)			NON-TECH (61)			Mann-Whitney	
		M_d	IQR	METRIC (AN)	M_d	IQR	METRIC (AN)	U	p
Q10	Dead – Alive	4	2	3	4	2	3.5	12	0.341
Q11	Stagnant – Lively	3	1		3	1			
Q12	Mechanical – Organic	2	2		3	1			
Q13	Artificial – Lifelike	2	2		3	2			
Q14	Inert – Interactive	3	1		4	2			
Q15	Apathetic – Responsive	4	2		4	2			
SERIES: LIKEABILITY									
GOAL G4 (QUESTION)	SEMANTIC ITEM	BY PROFESSIONAL BACKGROUND							
		TECH (52)			NON-TECH (61)			Mann-Whitney	
		M_d	IQR	METRIC (LK)	M_d	IQR	METRIC (LK)	U	p
Q16	Dislike – Like	4	2	5	4	1	5	10.5	0.699
Q17	Unfriendly – Friendly	5	1		5	1			
Q18	Unkind – Kind	5	1		5	1			
Q19	Unpleasant – Pleasant	5	1		5	1			
Q20	Awful – Nice	4.5	1		5	1			
SERIES: PERCEIVED INTELLIGENCE									
GOAL G5 (QUESTION)	SEMANTIC ITEM	BY PROFESSIONAL BACKGROUND							
		TECH (52)			NON-TECH (61)			Mann-Whitney	
		M_d	IQR	METRIC (PI)	M_d	IQR	METRIC (PI)	U	p
Q21	Incompetent – Competent	4	1	4	5	2	5	5	0.067
Q22	Ignorant – Knowledgeable	4	1		4	1			
Q23	Irresponsible – Responsible	4	1.25		5	1			
Q24	Unintelligent – Intelligent	4	2		4	1			
Q25	Foolish – Sensible	4	2		5	1			

test indicated no statistically significant difference between the groups ($U = 12$, $p = 0.341$, $r = 0.33$).

6.3.3.4 G4 – Perceived Likeability

To address G4, which aims to evaluate the FRED robot with respect to the Likeability dimension based on user perception, the LK metric was computed from responses to questions Q16 to Q20, corresponding to the Likeability dimension of the GQS. The overall LK metric resulted in $M_d = 5$ and $IQR = 0$.

Group Comparison by Gender. For this dimension, the Male group presented $M_d = 4$ and $IQR = 0$, whereas the Female group showed $M_d = 5$ and $IQR = 0$. The values resulting from statistical test were $U = 0$, $p = 0.003$ and $r = 0.95$, indicating a statistically significant difference when comparing the groups.

Group Comparison by Professional Background. When comparing by background professional, both groups showed a median of $M_d = 5$, with $IQR = 0.5$ for the Tech group and $IQR = 0$ for the Non-Tech group. The statistical test returned $U = 10.5$, $p = 0.699$ and $r = 0.16$, indicating that there was no statistically significant difference when comparing the groups.

6.3.3.5 G5 – Perceived Intelligence

To address G5, the Perceived Intelligence dimension was assessed using the PI metric, derived from participants' responses to questions Q21 to Q25 of the GQS. The overall PI metric resulted in $M_d = 4$ and $IQR = 0$.

Group Comparison by Gender. For this dimension, both groups obtained identical values, with $M_d = 4$ and $IQR = 0$. The values from statistical test were $U = 10$, $p = 0.424$ and $r = 0.20$, indicating that there was no statistically significant difference when comparing the groups.

Group Comparison by Professional Background. For the same dimension, the Tech group presented $M_d = 4$ and $IQR = 0$, whereas the Non-Tech group obtained $M_d = 5$

Table 29: Value of the PCA metric for all participants, for the groups (Male and Female), for the groups (Tech and Non-Tech) and the statistical values resulting from the Mann-Whitney test.

GOAL G6 (QUESTION)	METRIC	ALL		BY GENDER						BY PROFESSIONAL BACKGROUND					
		PARTICIPANTS		MALE		FEMALE		Mann-Whitney		TECH		NON-TECH		Mann-Whitney	
		(n = 113)		(n = 71)		(n = 42)				(n = 52)		(n = 61)			
		M_d	IQR	M_d	IQR	M_d	IQR	U	p	M_d	IQR	M_d	IQR	U	p
Q26	PCA	5	1	5	1	5	1	1423	0.637	5	1	5	1	1317	0.069

and $IQR = 1$. The statistical test resulted in $U = 5$, $p = 0.067$ and $r = 0.60$, indicating that there is no statistically significant difference in the comparison between the groups.

6.3.3.6 G6 – Perceived Child Acceptance

To address G6, the Perceived Child Acceptance was assessed using the PCA metric, derived from participants' responses to question Q26. The overall result presented $M_d = 5$ and $IQR = 1$.

Group Comparison by Gender. Responses to the statement "*Children would like to interact with FRED*" resulted in identical values for both groups, with $M_d = 5$ and $IQR = 1$. The statistical test returned $U = 1423$, $p = 0.637$ and $r = 0.04$, indicating no statistically significant difference when comparing the groups.

Group Comparison by Professional Background. The Tech and Non-Tech groups both presented $M_d = 5$ and $IQR = 1$. The statistical test resulted in $U = 1317$, $p = 0.069$ and $r = 0.17$, indicating no statistically significant difference when comparing the groups.

6.3.3.7 Qualitative Analysis of Participant Feedback

To complement the quantitative assessment, participants' open-ended responses were analyzed qualitatively to provide additional insights into how they perceived the FRED robot's affective expressions and characteristics, complementing the quantitative results obtained through metrics such as the PEA and dimensions of GQS. A total of 52 out of 113 participants provided written feedback. The responses were organized into units of analysis, with each unit corresponding to an individual idea expressed in an open-ended response. Responses containing more than one idea were segmented accordingly, whereas those containing a single idea were retained as a single unit of analysis. This process resulted in 74 units.

Table 30: Categories and subcategories defined based on participants' open-ended comments regarding FRED's affective expression and its characteristics according to the GQS dimensions.

Category	Subcategory	Representative Units of Analysis
Appearance	Shape (overall aesthetics) Anthropomorphic features Color and Style	"I found its appearance great" "A very mechanical model" "I suggest a more rounded structure, as it currently resembles Lego figures"
Facial Expressions	Clarity of emotions Intensity of expressions Eye design and Gaze	"Its facial reactions are very creative" "The eyes when sad are hard to interpret" "An improvement in the expression of sadness would be interesting"
Body Movement	Leg and Foot movements Absence of gestures (Arms)	"Liked the little legs stretching with happiness" "I didn't quite understand the foot movements" "It would be good to move the arms too"
Speech and Voice	Prosody Naturalness of speech Response latency (TTS)	"Very nice dialogue system though" "The intonation does not match the emotion" "Response speed could be improved"

Units of analysis addressing similar aspects of the robot were then grouped into subcategories, which were subsequently organized into four broader categories: (1) Appearance, including the subcategories shape, anthropomorphism, and color and style; (2) Facial Expressions, including the subcategories clarity of emotions, intensity of expressions, eye design, and gaze; (3) Body Movement, including the subcategories leg and foot movements and the absence of gestures (e.g., arms); and (4) Speech and Voice, including the subcategories prosody (rhythm, intonation, accent, and volume), naturalness of speech, and response latency (Text-to-Speech generation). Each unit was assigned to a single feedback type and a single subcategory. Representative units of analysis were selected to illustrate the resulting categories, and the frequency of positive feedback (35.14%), negative feedback (13.51%), and improvement suggestions (51.35%) units was calculated for each broader category.

Table 30 provides an overview of the categories, subcategories, example quotations of units of analysis classified as positive feedback, negative feedback, and improvement suggestions.

Appearance. Appearance was the most frequently mentioned category, accounting for 27 units of analysis (36.49% of the 74 units). Among these, 15 corresponded to positive feedback. For instance, P2 and P44 stated, respectively, *"I found its appearance great"* and *"Regarding appearance, I found the robots quite friendly and even charismatic."*

Negative feedback was limited 2 units of analysis, referring to aspects such as a *"very mechanical model"* (P10) and a *"very boxy appearance"* (P106).

The number of units of analysis for improvement suggestions was 10. For example, P40 stated, *"The robot could have a more humanoid aspect."* Regarding shape, P1 and P54 commented, respectively, *"Regarding its appearance, I suggest more rounded shapes in its structure"* and *"Perhaps a more rounded aspect would be friendlier."* Additional suggestions included *"I also believe that if the robots had hair, children would like them even more"* (P8) and *"Using clothing with fabrics on the robot's body"* (P1).

Facial Expressions. The Facial Expressions category accounted for 15 units of analysis (20.27% of the 74 units). Among these, four corresponded to positive feedback that included statements such as *"Its facial reactions are very creative"* (P36) and *"I think FRED's appearance is amazing, as well as its ways of expression"* (P88).

Negative feedback (4 units) included units such as *"I found the expression of anger to be very exaggerated"* (P75), *"I found the robots' expression of anger to be exaggerated, scary!"* (P79), and *"The expressions are very mechanical; it is almost imperceptible to notice happiness or sadness in its face"* (P94). Participant P111 commented, *"I feel the way the eyes are set during sadness is a bit difficult to interpret on its own as sadness."*

Improvement suggestions (seven units), included statements such as *"I cannot suggest exactly how, but an improvement in the sadness expression would be interesting"* (P20), *"The expression of sadness could be improved"* (P55), and *"Improve the display screen of the expressions, in terms of resolution"* (P1).

Body Movement. Body Movement was associated with 18 units of analysis (24.32% of the 74 units). Positive feedback, with three units, included statements such as *"I liked the little legs stretching with happiness"* (P78) and *"Demonstrated great synchronicity between speech and facial and body expressions"* (P91).

Negative feedback (two units) included *"The body expressions are average"* (P60) and *"I didn't quite understand the foot movements"* (P36).

Improvement suggestions (13 units) included statements such as *"I think it would be a good idea to include arm movements"* (P113), *"More body expressions, to further intensify facial expressions"* (P90), and *"More limb movements to express emotions"* (P106). Additional comments included *"There could be movements during the dialogue"* (P17).

Speech and Voice. The Speech and Voice category comprised 14 units (18.92% of the 74 units). Positive feedback (four units) included *"Very nice dialogue system though"* (P111) and *"It showed great synchronization between speech and facial and body expressions"* (P91).

Negative feedback, with two units, included *"The robots' voices don't express any emotion"* (P60) and *"The intonation of what is said does not correspond to the emotion"* (P66).

Improvement suggestions (eight units) included statements such as *"More fluidity in the dialogue"* (P18), *"Convey anger in the sound"* (P78), and *"Perhaps it would be interesting for the robots to speak in a more natural way"* (P44). Additional comments included *"Improve the robot's reaction time"* (P37) and *"The speed of the responses could be improved"* (P21).

Figure 53 presents a bar chart showing the frequencies of positive feedback, negative feedback, and improvement suggestions for each of the defined categories across 74 units of analysis.

Figure 54 presents a word cloud generated exclusively from participants' positive feedback, highlighting the most frequent adjectives used to describe favorable impressions of the robot.

6.3.4 Discussion

6.3.4.1 Affective Expression Capability of FRED

The results obtained based on the values of the PEA metric showed evidence that participants were able to identify more accurately, responding in the expected way, to emotions expressing happiness and anger, with 84.81% and 76.4%, respectively. The high percentage of correct answers in identifying happiness suggests that this is the emotion most effectively conveyed by the FRED robot. In the evaluation of the happiness emotion, more specifically, in Block 1, although 58.4% of the participants identified the absence

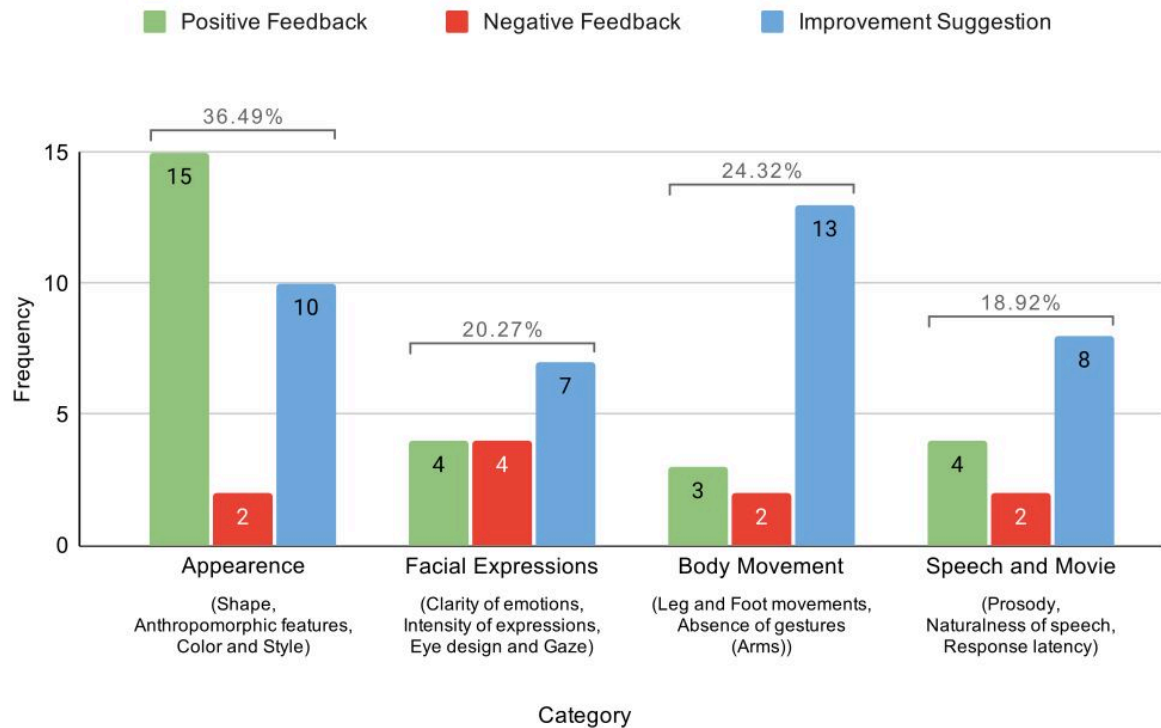


Figure 53: Frequency of positive feedback, negative feedback, and improvement suggestions for each of the categories defined for 74 units of analysis. The percentages of the number of units of analysis for each category are also presented.



Figure 54: Word cloud based on participants' positive feedback, highlighting the most frequent terms (adjectives) used to describe favorable impressions of the robot. Word size reflects frequency.

of happiness in the expected way, 31.9% of them identified the robot with "Mild Happiness". The qualitative analysis suggests that the robot's general appearance contributes to a positive perception from the user's perspective. Comments related to the theme of Appearance, such as *"I found its appearance great"* and *"Regarding appearance, I found the robots quite friendly and even charismatic,"* indicate that FRED's naturally friendly design could explain why some participants perceived a level of happiness even in a state of neutrality. These indications appear in the word clouds generated from adjectives found in the feedback analysis units, with words such as "friendly," "good," "great," and "charismatic" appearing most frequently.

Although the FRED robot achieved a high PEA for the anger emotion (76.4%), the qualitative analysis reveals that the expression is perceived as *"very exaggerated"* and *"scary"* by some participants. This discrepancy can be explained by the lack of synchrony between facial expression and tone of voice. Comments in the Voice and speech theme, such as *"The intonation of what is said does not correspond to the emotion. It is always the same intonation,"* suggest that the lack of prosody in the speech can lead users to interpret the visual expression (clearly one of anger) as exaggerated or out of context, affecting the overall experience.

The emotions with the lowest PEA values were surprise and sadness. The surprise emotion could be better identified by the participants, with 62.68% and the sadness emotion obtained the lowest value, with 49.71%.

For the evaluation of the sadness emotion, an issue was observed that may have exerted some influence on the participant's perception. The issue is related to the design of the robot's eyes for the sadness expression. The poor performance in the perception of this emotion is corroborated by the qualitative analysis. Participants' comments indicate that the robot's visual representation of sadness was not clear enough. For example, the category Facial Expressions, specifically the subcategory *eye design and gaze*, elicited feedback such as *"I find the way the eyes are positioned during sadness a bit difficult to interpret as sadness on its own."* Some participants left the following suggestions regarding the expression of sadness: *"The expression in the emotion of sadness could be improved"*, *"An improvement in the expression of sadness would be interesting"*, *"I would suggest eyes point downwards, like (U U) could possibly accentuate the feeling of sadness."* Other participants left the following comments: *"Very imperceptible to notice happiness or sadness in its expression"*, *"I feel the way the eyes are set during sadness is a bit difficult to interpret on its own as a sadness"*. This feedback suggests that FRED's eye

design may be a key factor in the poor perception of sadness.

The results also showed that, in some situations, although participants were able to identify the emotion displayed by the robot FRED as expected, they did not accurately identify the expected intensity. Participants' difficulty in accurately assessing the intensity of emotions, such as the tendency to overestimate sadness for "Intense Sadness" rather than "Mild Sadness," suggests a limitation in FRED's modulation of expressions. This perception of the quantitative data is directly supported by the qualitative feedback on the Body Movement category. Participants actively suggested the need for more movement, with comments such as *"More body expressions, to further intensify the facial expressions"* and *"I think including arm movements would be a good idea, since it's easier to understand emotions with body expressions."* This indicates that the robot, with its limited range of movements, cannot communicate nuances of intensity, leading users to interpret existing expressions more exaggeratedly. This occurred with the sadness emotion, in Blocks 3 and 5, where a higher percentage of participants (54.9% and 55.8%, respectively) indicated an intensity of "Intense Sadness" when the response "Mild Sadness" was expected (39.8% and 37.2%, respectively). These values influenced the result of the PEA metric for the sadness emotion and showed evidence that the robot's poses may not be sufficient to modulate the intensity of the emotion demonstrated by the robot, requiring the addition of more elements of non-verbal communication, such as arm movements, for example.

In order to maintain continuity in the context of the dialogue between the robots, between the video blocks, some of these blocks began with the robot demonstrating the emotion of the end of the previous block and this may have generated a misunderstanding in the expected interpretation of the emotions in the respective blocks. This may have occurred during the evaluation of Block 4. In this block, the video begins with the robot expressing the last emotion from the previous block, "Intense Surprise," but it is not repeated throughout the block, unlike what happened in the previous block. In this case, participants may not have considered the initial emotion of the block ("Intense Surprise") since it does not occur again in the block. This may have been the reason why a larger proportion of participants (46.9%) reported not perceiving this emotion in the block.

The prevalence of words such as "great," "good," and "friendly" in the word cloud shown in Figure 54 suggests that positive perceptions may be associated with the robot's empathetic and cordial behavior, indicating that the interaction went well and the robot fulfilled its role. This finding suggests that participants' positive evaluations are more closely linked to the social qualities perceived in the robot than to specific functional

attributes. Qualitative results indicate that participants rated FRED positively regarding its appearance and the interaction; however, several aspects, such as emotional clarity, bodily expressiveness, and vocal modulation, remain key areas for improvement.

Male and Female groups. Regarding gender, minor variations in central tendency were identified in specific cases, such as sadness in Block 3 (Male: $M_d = 2$, Female: $M_d = 1.5$) and surprise in Block 4 (Male: $M_d = 0$, Female: $M_d = 1$). However, these differences were not statistically significant (e.g., $p = 0.385$ and $p = 0.521$, respectively), indicating that gender did not significantly influence participants' perception of the robot's affective expressions.

Tech and Non-Tech groups. A brief discussion regarding the questions that showed a statistically significant difference in the comparison between the responses of the Tech and Non-Tech groups will be presented.

As shown in Section 6.3.3.1, despite the statistical difference indicated by the p value in seven answers, in six of them, the participant's answers have the same value for the medians. This behavior may occur because the nonparametric Mann-Whitney test does not take into account only the median values, but also how the data are distributed (HART, 2001; CONROY, 2012). The distributions of the answers for the emotions and blocks where there was a statistical difference between the answers of the two groups, can be better observed with the help of the graphics that are shown in Figure 55.

As can be seen in Figure 55, for the question that assessed the emotion "sadness", in Block 1, although the medians of the answers for both groups were equal to 0 ("No Sadness"), indicating the correct perception for the FRED expression for the absence of sadness, it can be seen that there is a greater concentration of 0 answers for the Tech group, while for the Non-Tech group, the answers spread to options 1 ("Mild Happiness") and 2 ("Intense Happiness"). As can be seen, in percentage numbers, in Figure 56, approximately 16% more participants in the Tech group responded the expected answer for this question.

For the questions related to the assessment of the emotion "anger", as can be seen in Figure 55, the distribution of data for both groups followed a shape similar to the previous case. The graphics for the Tech group present a wider base, around the value 0 ("No Anger") for blocks 1, 2 and 3. From Figure 57 it can be observed that, for the three blocks, at least 16% more participants in the Tech group answered as expected. While at

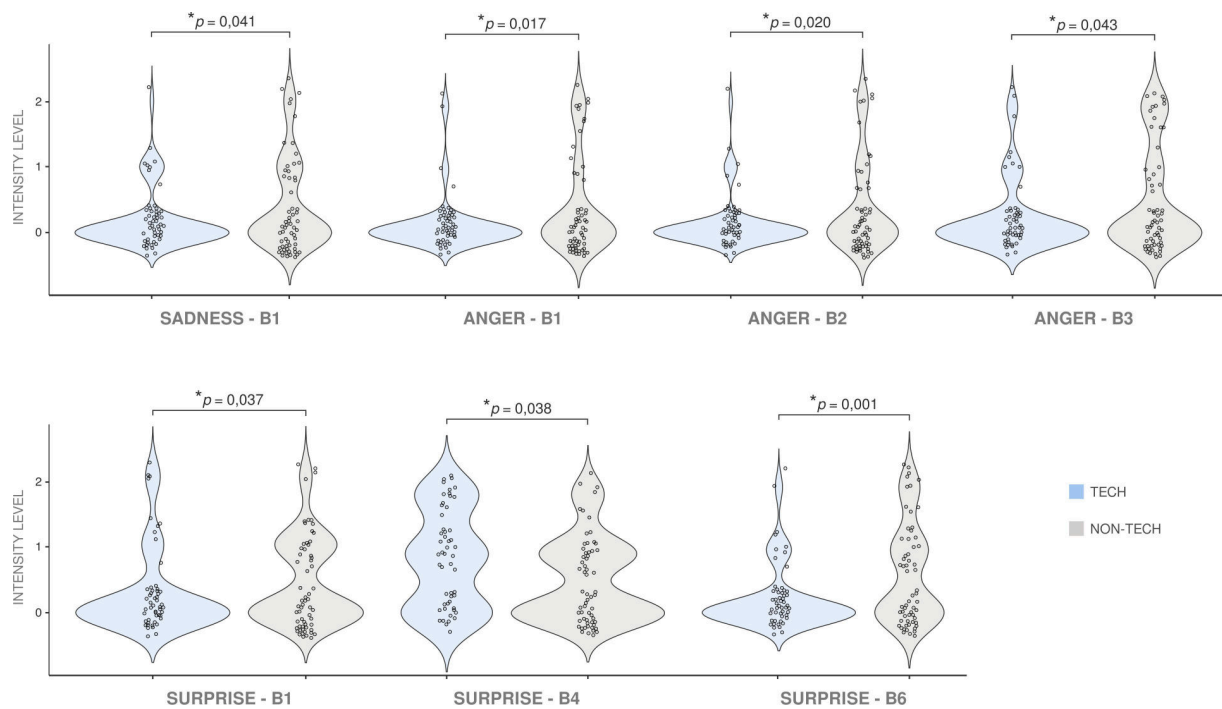


Figure 55: Graphics showing the distributions of answers regarding emotions and blocks in which the Mann-Whitney test returned $p < 0.05$, indicating a statistical difference between the answers of the Tech and Non-Tech groups.

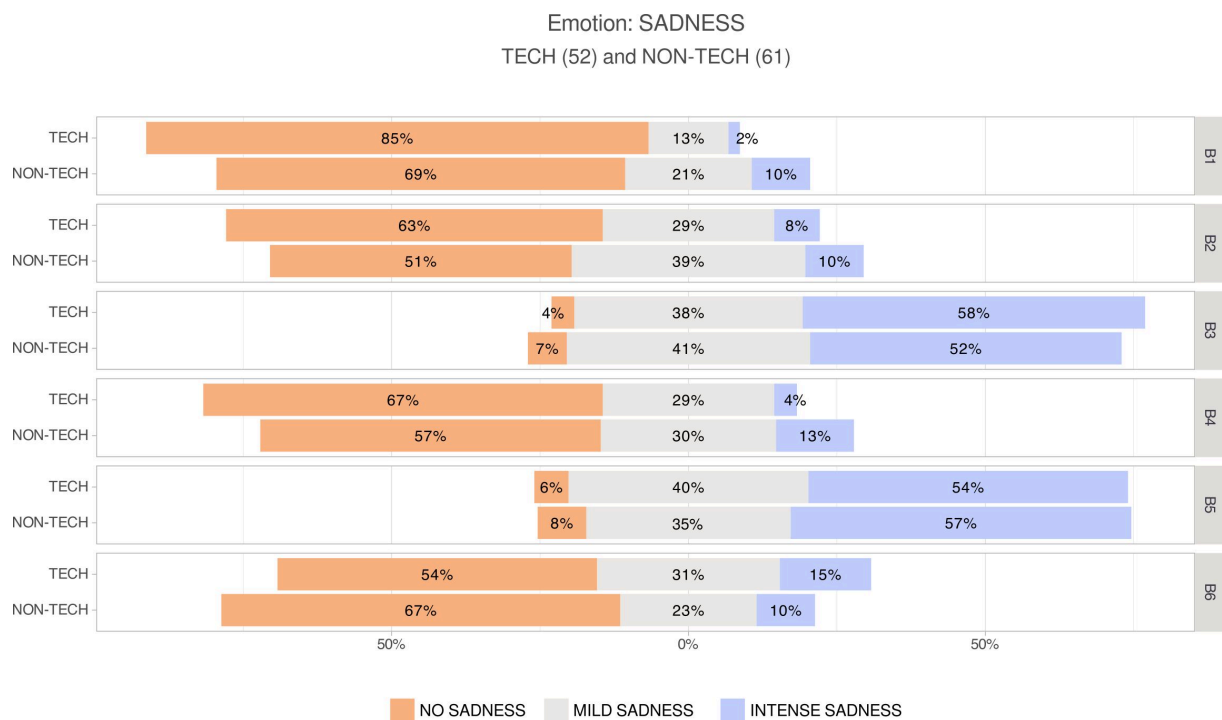


Figure 56: Percentages for answers from technology and non-technology participants to question Q2 ("Did you perceive when the robot expressed sadness?") for each of the six blocks.

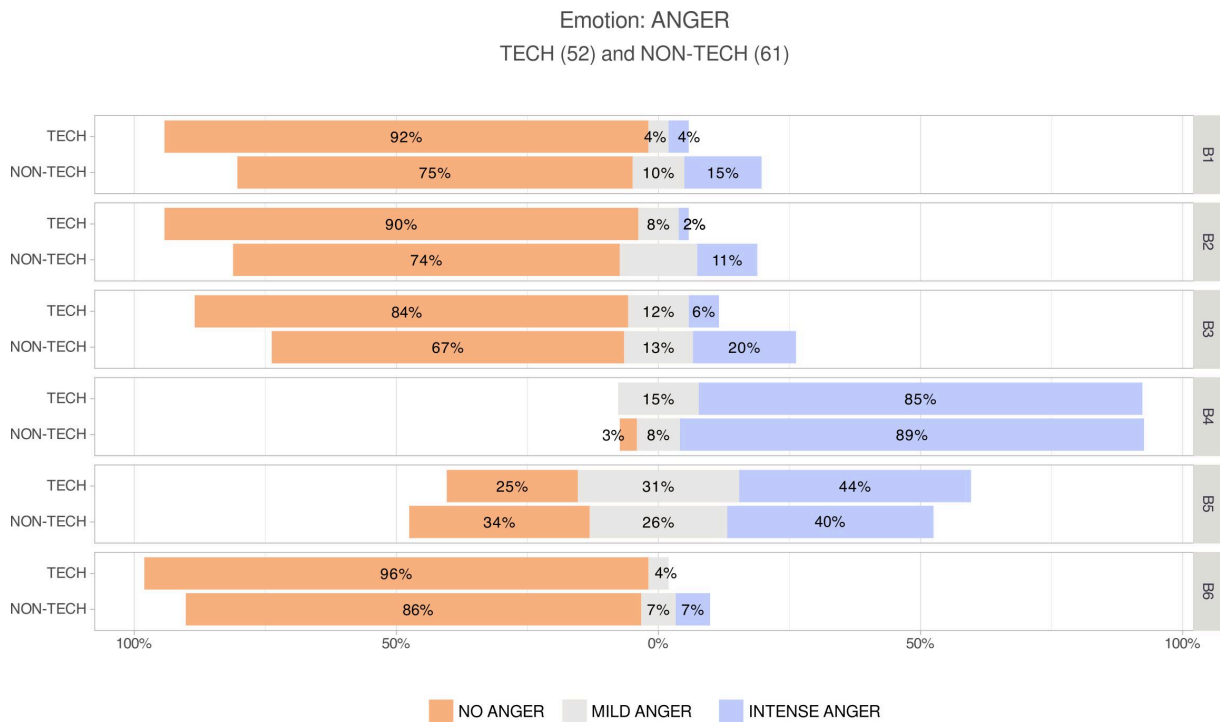


Figure 57: Percentages for answers from technology and non-technology participants to question Q3 (*"Did you perceive when the robot expressed anger?"*) for each of the six blocks.

least 23% of the answers from the Non-Tech group were distributed among the answers with options 1 ("Mild Anger") and 2 ("Intense Anger").

For the question related to the evaluation of the emotion "surprise", in Block 1, as can be seen in Figure 55, the shape for the distribution of the Tech group continues with characteristics that resemble the previous distributions, with a percentage greater than 80% for the expected answer, 0 ("No Surprise"), as can be seen in Figure 58. For the Non-Tech group, the percentage for the expected answer was only 61%, 19% less than that of the Tech group. The Tech group continues with a similar shape for the distribution of answers in the evaluation of the emotion "surprise", in Block 6. The group presented a percentage of approximately 83% of expected answers, against 54% for the Non-Tech group, a difference of 29%. In this Block 6, the answers of the Non-Tech group spread even further to options 1 ("Mild Surprise") and 2 ("Intense Surprise").

As shown in Section 6.3.3.1, in the evaluation of the emotion "surprise", in Block 4, 46.9% of the participants (considering all 113 who participated in the evaluation) did not respond with the expected answer, indicating the answer 0 ("No Surprise"). However, in this comparison between groups, it can be seen that the Tech group, unlike the Non-Tech group, presented in Figure 55, a distribution with a shape very different from the

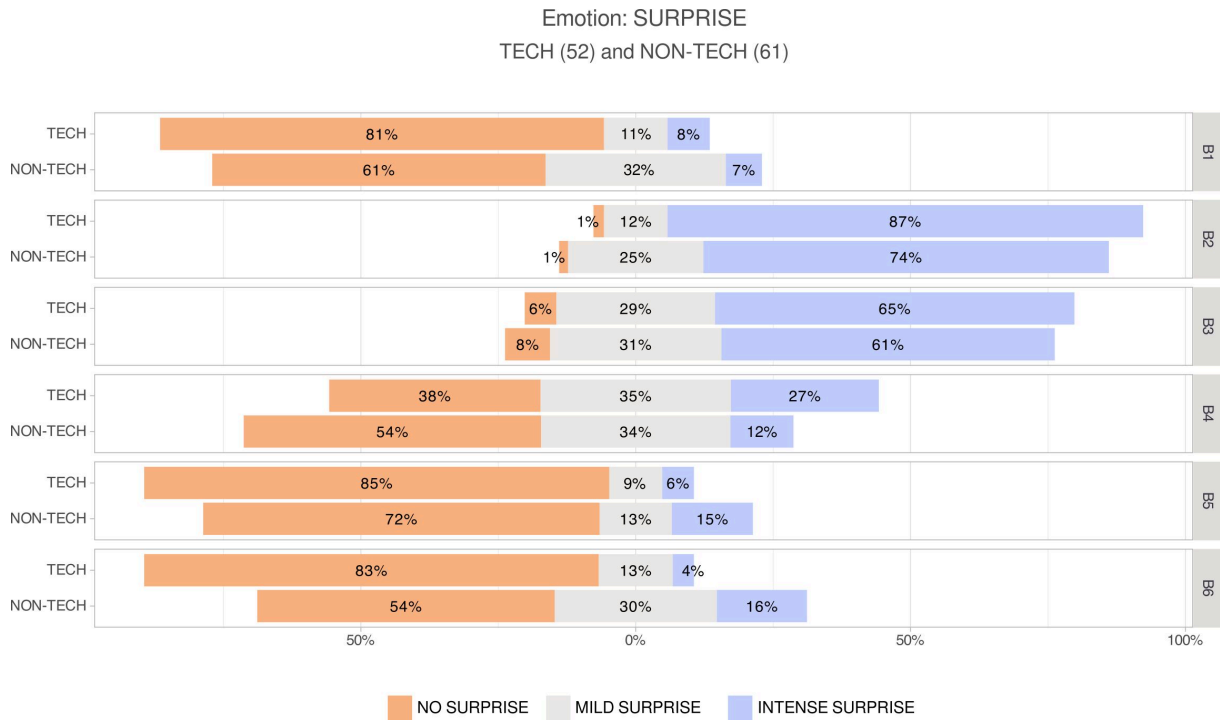


Figure 58: Percentages for answers from technology and non-technology participants to question Q4 (*"Did you perceive when the robot expressed surprise?"*) for each of the six blocks.

previous ones, with the participant's answers being distributed between the options 0, 1, not making it so clear where the data is most concentrated. The answers of the Tech group presented a median with value $Md = 1$ and $IQR = 2$. Although it was not the expected answer, which would be 2 ("Intense Surprise"), 62% of the participants in the group were able to identify the emotion "surprise" at some level of intensity in the robot, as can be seen in Figure 58.

From the analyzed data in this section, participants working in the technological area, compared to those working in non-technological areas, had a better perception of the affective expressions of the FRED robot. This suggests that individuals with experience in technology may be more familiar with robotic expressions, perhaps due to exposure to similar technologies or concepts in their professional environment.

6.3.4.2 Perceived Anthropomorphism

Based on the results presented in Section 6.3.3.2, the value obtained for the AT metric, associated with the Anthropomorphism dimension of the GQS, was equal to $Md = 3$. This was the lowest value among the values of the metrics defined for this work. Based on this value, we can conclude that the participants positioned themselves with neutrality,

demonstrating uncertainty in their perceptions about the anthropomorphic characteristics of FRED.

The p value resulting from the statistical test in the comparison between the groups (Male and Female) and (Tech and Non-Tech) was greater than 0.05, indicating that there is no statistically significant difference in the participants' perception of the robot in relation to the Anthropomorphism dimension, based on gender or professional background.

6.3.4.3 Perceived Animacy

Section 6.3.3.3 presented the results of the AN metric, associated with the GQS Animacy dimension. The value obtained for the metric was $Md = 3.5$. This was the second lowest result among the four metrics defined. From this value, we can conclude that the participants also positioned themselves neutrally in relation to the semantic items of the Animacy dimension, such as (Dead – Alive), (Mechanical – Organic), (Artificial – Lifelike) etc.

The p value resulting from the statistical test in the comparison between the groups (Male and Female) and (Technological and Non-Technical) was greater than 0.05, indicating that there is no statistically significant difference in the participants' perception of the FRED robot in relation to the Animacy dimension, based on gender or professional training.

6.3.4.4 Perceived Likeability

Regarding the LK metric, which is associated with the Likeability dimension of the GQS, Section 6.3.3.4 presented the value $Md = 5$. This metric was the one that obtained the highest value, the maximum value, in the participants' perception. Based on this value, we can conclude that the participants found FRED to be likeable, friendly, kind, nice, etc.

Observing the LK metric values for males and females to compare the groups, the metric value for the Male group was $Md = 4$ and for the Female group it was $Md = 5$. The statistical test returned the values $U = 0$ and $p = 0.003$. The value of p , less than 0.05, indicates that there is a statistically significant difference between the perceptions of the two groups. Based on this, we can conclude that gender influenced the participants' perception of FRED in relation to the Likeability dimension, indicating that females found FRED more likeable ($Md = 5$) than males ($Md = 4$).

For the Tech and Non-Tech groups, the LK metric value was the same for both groups, with the value $Md = 5$. The values returned from the statistical test were $U = 10.5$ and $p = 0.699$. Since the value of p is greater than 0.05, the statistical test showed that there is no statistically significant difference between the opinions of the two groups. Based on this information, we can conclude that the participants' professional background did not influence their perceptions of FRED in relation to the Likeability dimension, indicating that both groups (Tech and Non-Tech) found FRED likeable, friendly, kind, nice, etc.

6.3.4.5 Perceived Intelligence

Section 6.3.3.5 presented the results of the PI metric, associated with the Perceived Intelligence dimension of GQS. The value obtained for the metric was $Md = 4$. This was the second best result among the four metrics defined. Although it is not the maximum value, we can conclude that the participants perceived FRED as partially competent, partially knowledgeable, partially responsible, partially intelligent and partially sensitive.

For the Male and Female groups, the PI metric values for both groups were the same, $Md = 4$. The statistical test returned the values $U = 10$ and $p = 0.424$. The value of p , greater than 0.05, indicates that there was no statistically significant difference between the perceptions of the two groups. Based on these values, we can conclude that gender did not influence the participants' perception, indicating that both groups perceived FRED as partially competent, partially knowledgeable, partially responsible, partially intelligent and partially sensitive.

For the Tech and Non-Tech groups, given that the p value, resulting from the statistical test was greater than 0.05, indicating the absence of a statistically significant difference between the metrics of the two groups, it can be concluded, based on these values, that the participants' professional backgrounds did not influence their perceptions of the FRED robot regarding the Perceived Intelligence dimension.

6.3.4.6 Perceived Child Acceptance

Based on the values presented in Section 6.3.3.6, the result ($Md = 5$) for the Perceived Child Acceptance (PCA) metric indicated that participants strongly agreed that "*Children would like to interact with FRED robot*". This result is in line with some comments and suggestions left by participants: "*I love FRED and I think kids will love it!*", "*I think FRED looks amazing, as well as his ways of expression. There's no way a child wouldn't*

like it and interact with it", "Looks great! I can see many scenarios where FRED would be useful and entertaining, especially for kids".

The p value resulting from the statistical test in the comparison between the groups (Male and Female) and (Tech and Non-Tech) was greater than 0.05, indicating that there is no statistically significant difference in the answers based on gender or professional background.

6.3.4.7 Semantic Item Exploration

As in the work of [Schulz et al. \(2019\)](#), this section discusses the results of the comparisons between the participants' answers focused on the semantic items of each of the GQS dimensions selected for this work. The results of the exploration of these items may point to possible specific characteristics of the FRED robot that should be improved in future studies.

First, a discussion will be made regarding the perceptions of each user group (Male and Female, and Tech and Non-Tech) based only on the observation of the median values presented in the graphics in Figure 59. Then, the results of the statistical comparison of the median answers by gender will be discussed. Finally, the results based on the statistical comparison of the groups by professional background will be discussed. Figure 59 contains two radar charts. This type of chart makes it easier to identify patterns and differences between categories. They are showing the median values of the participants' answers (on a five-point Likert scale) for each of the questions (Q5 to Q25) associated with each of the semantic items of the GQS dimensions selected for the evaluation of the FRED robot (Anthropomorphism, Animacy, Likeability and Perceived Intelligence). The graphic on the left shows the values by gender and the graph on the right shows the values by professional background.

Comparison between answers from male and female participants Observing the graphic for the groups separated by gender, presented in Figure 59 (left), it can be seen that the median values of the females' answers were equal to or greater than those of the males, with the exception of Q6 (Machine Like – Human Like), which was equal to 2, while the median value for the males was equal to 3. This information suggests that females, in general, had a better impression of FRED, especially in the Likeability dimension, where females evaluated the robot with the maximum value ($Md = 5$), for all items. Another evaluation with maximum value ($Md = 5$), from the Females group, was for the semantic

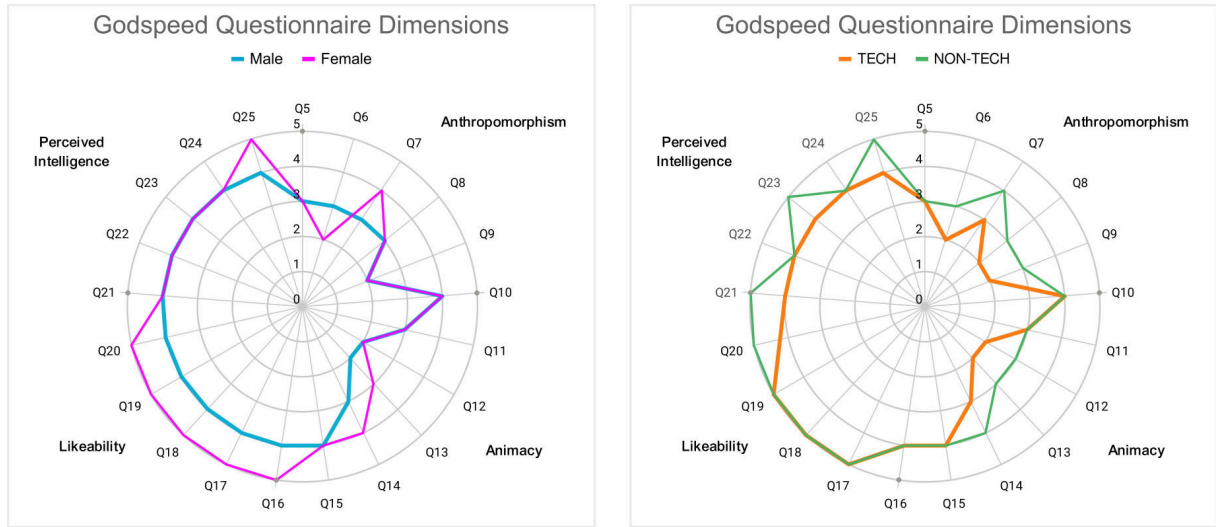


Figure 59: Radar charts showing median values calculated from participant answers for each group, Male and Female (left) and Technological and Non-Technological (right).

item (Foolish – Sensible), referring to question Q25, of the Perceived Intelligence dimension. It can also be seen that the maximum median value for the Males group evaluations, for the Perceived Intelligence dimension, was equal to 4. The evaluations with the lowest median values, for both genders, were in the Anthropomorphism and Animacy dimensions. Females, for the item (Machine Like – Human Like), associated with question Q6, perceived the FRED as partially "Machinelike", with a median value of 2 and for question Q9, semantic item, (Moving rigidly – Moving elegantly), like males, perceived the robot partially "Moving rigidly", with a median value of 2. For question Q12, item (Mechanical – Organic), the median values of the robot evaluation, for both genders, were equal to 2, that is, males and females, in relation to the Animacy dimension, perceived the FRED partially "Mechanical". For question Q13, item (Artificial – Lifelike), the males evaluated FRED's movement as partially "Artificial", with a median value of 2. FRED did not obtain any evaluation with a minimum median value (1).

After the discussion on the answer values by gender, presented in the graphic (on the left) of Figure 59, which was based only on the median values of the participants' answers, the nonparametric Mann-Whitney statistical test was applied to these answers, for each of the semantic items, of the four dimensions of the GQS used in this work, in order to verify if there is a statistically significant difference between the perceptions of the two groups. Table 31 presents the median values and interquartile ranges for the groups, Male and Female, in addition to the values resulting from the Mann-Whitney statistical test with the values of U and p for each of the groups.

As can be seen in Table 31 the statistical tests returned a p value lower than 0.05

for seven semantic items, two related to the Animacy dimension and five related to the Likeability dimension. The median values for the semantic item (Inert – Interactive), associated with question Q14, for the Male and Female groups, were 3 and 4. Based on the values of the statistical test for this GQS item ($U = 1044$ and $p = 0.006$), it can be concluded that females found FRED more interactive than males.

For the item (Apathetic – Responsive), associated with question Q15, the median values were equal to 4, for both groups. The results of the statistical test for this item were $U = 1112.5$ and $p = 0.018$, indicating that there is a statistically significant difference between the answers of the two groups. Since the Mann-Whitney test takes into account the distributions of the answers, it is possible to see, through the violin graphic (on the left) in Figure 60 that, although the medians were equal to 4, the answers of the females are much more concentrated near the values 4 and 5, while those of the males are spread across lower values, including the minimum value, 1. This visual information can be confirmed numerically through the data presented in the graphic (on the right) of the same figure. As can be seen, 36% of the females perceived FRED as being "Responsive", against 27% of the males. Furthermore, we can see that 19% of the males demonstrated a more negative perception of FRED with 4% of the medians equal to 1 ("Apathetic") and 15% equal to 2. This information may suggest that the females found the robot more responsive than the males.

For the Likeability dimension (Q16 to Q20), the medians of the males' answers were all equal to 4 and those of the females were equal to 5. The p values for all semantic items of this dimension, associated with questions Q16 to Q20, were less than 0.05, indicating that there is a statistically significant difference between the answers of the two groups. From this, it can be concluded that the females liked FRED more, finding it more likeable, friendly, kindful, pleasant and nice than the males.

Table 31: Values of the medians of the participants' answers for each semantic item, for each of the groups (Male and Female) and (Tech and Non-Tech, and the statistical values resulting from the Mann-Whitney test.

METRIC	QUESTION	SEMANTIC ITEM	BY GENDER						BY PROFESSIONAL BACKGROUND					
			MALE (n = 71)		FEMALE (n = 42)		Mann Whitney		TECH (n = 52)		NON-TECH (n = 61)		Mann Whitney	
			Md	IQR	Md	IQR	U	p	Md	IQR	Md	IQR	U	p
AT	Q5	Fake – Natural	3	2	3	1	1336.5	0.343	3	2	3	1	1283.5	0.072
	Q6	Machinelike – Humanlike	3	1	2	2	1317.5	0.289	2	2	3	2	1216.5	0.028*
	Q7	Unconscious – Conscious	3	1	4	2	1256.5	0.153	3	1	4	2	1269	0.061
	Q8	Artificial – Lifelike	3	1,5	3	2	1450.5	0.807	2	2	3	2	1157.5	0.011*
	Q9	Moving rigidly – Moving elegantly	2	2	2	2	1475	0.924	2	1.25	3	2	1077.5	0.002*
AN	Q10	Dead – Alive	4	2	4	1,75	1172	0.05	4	2	4	2	1585	0.998
	Q11	Stagnant – Lively	3	1	3	1	1191.5	0.059	3	1	3	1	1503.5	0.616
	Q12	Mechanical – Organic	2	2	2	1	1405.5	0.601	2	2	3	1	1215	0.027*
	Q13	Artificial – Lifelike	2	1	3	2	1251.5	0.144	2	2	3	2	1095.5	0.004*
	Q14	Inert – Interactive	3	2	4	2	1044	0.006*	3	1	4	2	1534	0.759
	Q15	Apathetic – Responsive	4	2	4	1	1112.5	0.018*	4	2	4	2	1515.5	0.672
LK	Q16	Dislike – Like	4	2	5	1	1110.5	0.015*	4	2	4	1	1538	0.769
	Q17	Unfriendly – Friendly	4	1	5	1	1136.5	0.018*	5	1	5	1	1514	0.643
	Q18	Unkind – Kind	4	1	5	0,75	1052.5	0.003*	5	1	5	1	1454.5	0.395
	Q19	Unpleasant – Pleasant	4	1	5	0	1101	0.009*	5	1	5	1	1451	0.382
	Q20	Awful – Nice	4	1	5	1	1126	0.015*	4.5	1	5	1	1461	0.421
PI	Q21	Incompetent – Competent	4	2	4	1	1214	0.084	4	1	5	2	1211	0.023*
	Q22	Ignorant – Knowledgeable	4	2	4	1	1277.5	0.181	4	1	4	1	1135	0.006*
	Q23	Irresponsible – Responsible	4	2	4	1	1432.5	0.713	4	1.25	5	1	1050	0.001*
	Q24	Unintelligent – Intelligent	4	2	4	1,75	1346.5	0.368	4	2	4	1	1222.5	0.028*
	Q25	Foolish – Sensible	4	2	5	1	1254.5	0.132	4	2	5	1	1209.5	0.020*

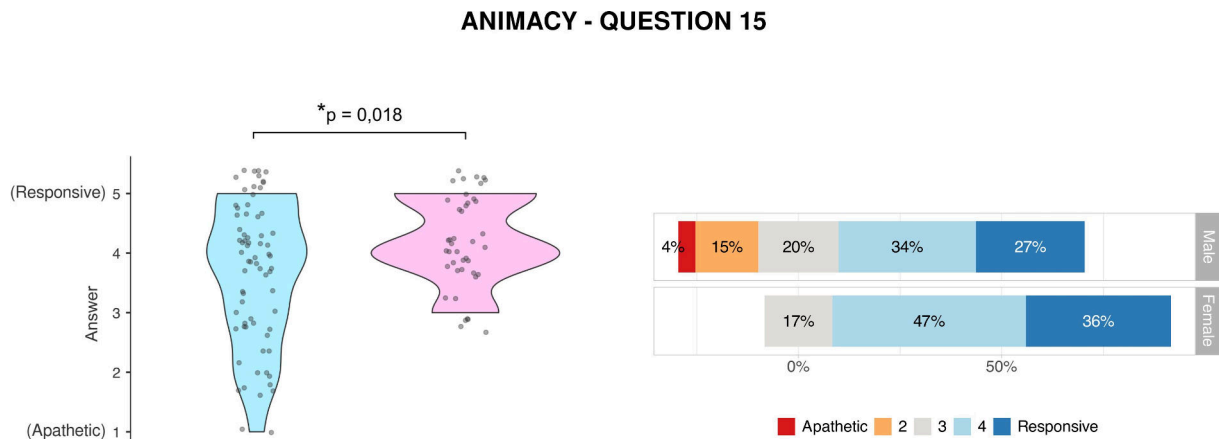


Figure 60: Two views (left and right) of the distributions of the medians of the answers of the participants in the Male and Female groups for the semantic item (Apathetic – Responsive).

Comparison between answers from technology and non-technology participants The graphic on the right of Figure 59 shows the median values for the answers of the Tech and Non-Tech groups. At first glance, it can be seen that the figure generated from the medians of the Tech group (polyline in orange) is contained in the polyline generated from the medians of the Non-Tech group (in green). Therefore, it is clear that the median values of the Non-Tech group, for all semantic items of the four dimensions, were greater than or equal to those of the Tech group. This suggests that, based only on the median values of both groups, people with non-technological professional background had, in general, a better impression of FRED. For the Non-Tech group, the Likeability dimension had the best evaluations for its items, with median values equal to 5, except for question Q16 (Dislike – Like) with median values equal to 4. For the Perceived Intelligence dimension, in addition to question Q25 (Foolish – Sensible), the median values for the Non-Tech group's answers, for questions Q21 (Incompetent – Competent) and Q23 (Irresponsible – Responsible), were also maximum ($Md = 5$). The lowest median values (for both groups) were also found in the Anthropomorphism and Animation dimensions. For the Tech group, the median values for questions Q6 (Machinelike – Humanlike), Q8 (Artificial – Lifelike) and Q9 (Moving rigidly – Moving elegantly), of the Anthropomorphism dimension, were the lowest ($Md = 2$). In addition to perceiving FRED partially as "Machinelike" and partially "Moving rigidly", participants in the Tech group also perceived it partially as "Artificial". The median values for the Non-Tech group, still for the Anthropomorphism dimension, for questions Q5 (Fake – Natural), Q6 (Machinelike – Humanlike), Q8 (Artificial – Lifelike) and Q9 (Moving rigidly – Moving elegantly), were equal to 3, suggesting

that participants in the Non-Tech group positioned themselves with neutrality in relation to these characteristics. Only for question Q7, semantic item (Unconscious – Conscious), the median value was equal to 4. The median values for the Tech group's answers, for the Animacy dimension, were also low ($Md = 2$). They were given for questions Q12 (Mechanical – Organic) and Q13 (Artificial – Realistic). Despite these questions with medians equal to 2, still for the Animacy dimension, questions Q10 (Dead – Alive), Q14 (Inert – Interactive) and Q15 (Apathetic – Responsive) obtained higher median values ($Md = 4$), suggesting that the participants of the Tech group perceived FRED, partially "Alive", partially "Interactive" and partially "Responsive".

After discussing the values of the answers by professional background, presented in the graph (right) of Figure 59, which was based only on the median values of the participants' answers, the nonparametric Mann-Whitney statistical test was applied to these answers, for each of the semantic items, of the four dimensions of the GQS used in this work, in order to verify whether there is a statistically significant difference between the perceptions of the two groups. Table 31 presents the median values and interquartile ranges for the Tech and Non-Tech groups, in addition to the values resulting from the Mann-Whitney statistical test, with the U and p values for each of the groups.

As can be seen in Table 31, the statistical test indicated a p value lower than 0.05 for ten semantic items, three related to the Anthropomorphism dimension, two to the Animacy dimension and five to the Perceived Intelligence dimension. For the Anthropomorphism dimension, the values returned by the statistical test for the semantic items (Machinelike – Humanlike), (Artificial – Lifelike) and (Moving rigidly – Moving elegantly), associated with questions Q6, Q8 and Q9, were ($U = 1216.5$ and $p = 0.028$), ($U = 1157.5$ and $p = 0.011$) and ($U = 1077.5$ and $p = 0.002$), respectively. Since all p values were lower than 0.05, there is a statistically significant difference between the perceptions of the two groups. For the Tech group, the median values were all equal to 2, and for the Non-Tech group, the median values were all equal to 3. Based on these values, it can be concluded that the Tech group had a less positive view of the robot, perceiving it partially as "Machinelike", partially "Artificial" and partially "Moving rigidly". The Non-Tech group, on the other hand, was neutral regarding these semantic items.

The same occurred with two semantic items related to the Animacy dimension. For the items (Mechanical – Organic) and (Artificial – Lifelike), associated with questions Q12 and Q13, the median values for the Tech and Non-Tech groups were 2 and 3, respectively, as can be seen in Table 31. In this case, the values returned by the statistical test were

$U = 1215$ and $p = 0.027$ and, $U = 1095.5$ and $p = 0.004$, respectively. With the p values lower than 0.05, indicating a statistically significant difference between the perceptions of the two groups, it can be concluded that the Tech group perceived FRED, in relation to the Animacy dimension, as partially "Mechanical" and partially "Artificial", while the Non-Tech group, placed itself in a position of neutrality in relation to its perceptions for these semantic items.

The values resulting from the statistical test applied to the medians of the answers to questions Q21, Q22, Q23, Q24 and Q25, of the Perceived Intelligence dimension were, respectively, ($U = 1211$ and $p = 0.023$), ($U = 1135$ and $p = 0.006$), ($U = 1050$ and $p = 0.001$), ($U = 1222.5$ and $p = 0.028$) and ($U = 1209.5$ and $p = 0.020$). As can be seen, the p values were all less than 0.05, indicating that there was a statistically significant difference between the perceptions of the two groups. For the semantic items (Incompetent – Competent), (Irresponsible – Responsible), (Foolish – Sensible) that are associated with questions Q21, Q23 and Q25, the median values for the Tech group were all equal to 4, and for the Non-Tech group, the median values were all equal to 5. Therefore, it can be concluded that the Tech group perceived FRED as being partially "Competent", partially "Responsible" and partially "Sensible", while the Non-Tech group, in fact, recognizes FRED as "Competent", "Responsible" and "Sensible".

The median values for the items (Ignorant – Knowledgeable) and (Unintelligent – Intelligent), which are associated with questions Q22 and Q24, were all equal to 4 for both groups. Although the median values were equal, the statistical test indicated that there was a statistically significant difference in the perceptions of the two groups. This may occur because, as previously mentioned, the Mann-Whitney test takes into account the distribution of the data. Observing the violin graph (on the left) in Figure 61, it is possible to get an idea of the distribution of the answers of the two groups for question Q22. As can be seen in the image, the distributions of the answers for the two groups follow a similar format, with the values being distributed in decreasing order, from the value 4 to the value 1. However, the Non-Tech group seems to have a greater concentration of answers with the maximum value ($Md = 5$) in relation to the Tech group. This visual observation can be transformed into numbers through the values presented in the graphic on the right, in the same figure. The percentage values prove a decreasing distribution of the medians from the value 4 (group Tech) and the difference between the number of medians with the value 5 between two groups. At least 80% of the participants in the Non-Tech group perceived FRED as partially "Knowledgeable", while for the Tech group, it was 63%. In addition, 46% of the participants in the Non-Tech group perceived the

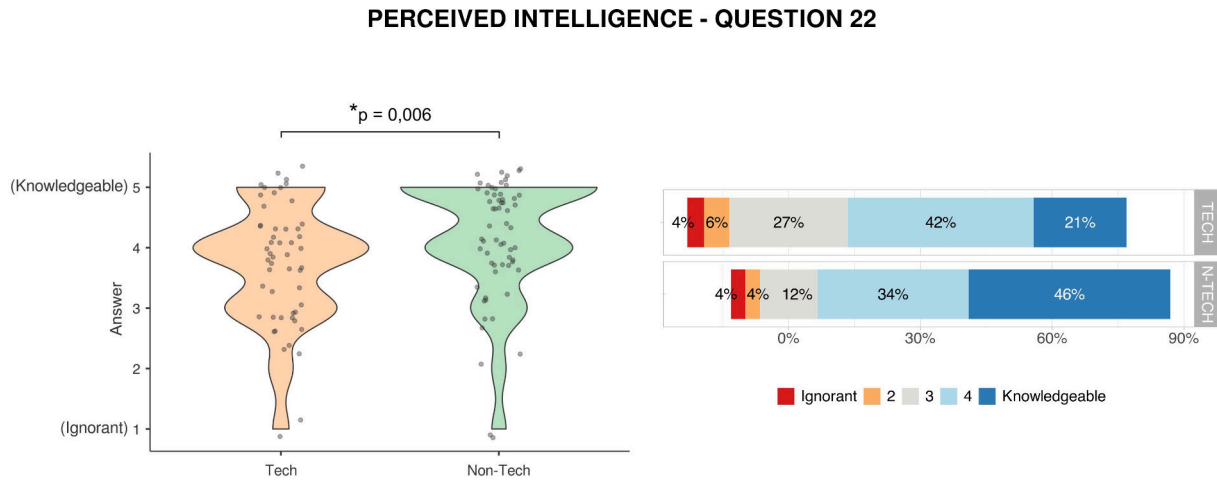


Figure 61: Two views (left and right) of the distributions of the medians of the answers of the participants in the Tech and Non-Tech groups for the semantic item (Ignorant – Knowledgeable).

robot as "Knowledgeable", more than double the participants in the Tech group, 21%. These data show evidence that the participants in the Non-Tech group perceived FRED as more "Knowledgeable" than the participants in the Tech group.

The same analysis can be made in relation to question Q24, associated with the semantic item (Unintelligent – Intelligent). By observing the violin graphic (on the left) in Figure 62 it is possible to see the distribution of answers from participants in both groups. For the Tech group, it can be seen that answer values are distributed in decreasing order, from value 5 to value 1, while for the Non-Tech group, a decreasing distribution can be seen, with a minimum value of 2. It can also be seen that for values 4 and 5, in the Non-Tech group, there is a greater concentration of answers. Through the graphic on the right, in the same figure, it is possible to see, in fact, that there is a greater concentration of answers 4 and 5 (78%) for the Non-Tech group, against 60% for the Tech group. Furthermore, 48% of the participants in the Non-Tech group, against 31% of the participants in the Tech group, perceived FRED as "Intelligent". These data show evidence that the Non-Tech group may have had a better perception of FRED for this semantic item.

6.3.5 Conclusion

This study aimed to evaluate the affective expressions of the open-source robotics platform FRED. For this purpose, a structured evaluation was carried out according to the GQM paradigm. The affective expressions of the robot and its characteristics based on the dimensions Anthropomorphism, Animacy, Likeability and Perceived Intelligence that are

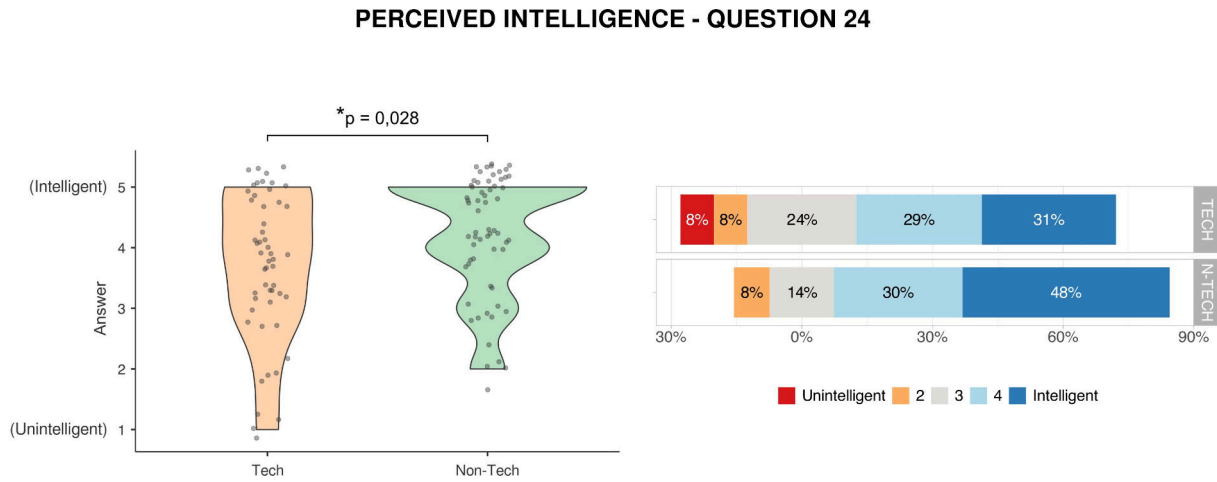


Figure 62: Two views (left and right) of the distributions of the medians of the answers of the participants in the Tech and Non-Tech groups for the semantic item (Intelligent – Unintelligent).

part of the GQS, were analyzed by 113 participants from different areas, such as: health, education, computer science and humanities.

To achieve Goal G1 (*"Analyze FRED's nonverbal communication capability with the purpose of evaluating its affective expressions, based on the user perception"*) participants evaluated FRED's ability to express happiness, sadness, anger, and surprise through its nonverbal communication elements, using the PEA (Percentage of Expected Responses) metric defined for this work. At the end of the evaluation stage, Goal G1 was achieved, showing that participants identified the emotions "happiness" and "anger", with values for the PEA metric equal to 84.81% and 76.4%, respectively, and identified the emotions "surprise" and "sadness", with percentages equal to 62.68% and 49.71%, respectively.

To achieve Goal G2 (*"Analyze the FRED robot with the purpose of evaluating its characteristics related to the Anthropomorphism scale, based on user perception"*), participants evaluated FRED according to the Anthropomorphism dimension that was associated with the AT metric. Goal G2 was achieved and participants were neutral regarding their perceptions of FRED's anthropomorphic characteristics. Furthermore, gender and professional background had no influence on participants' responses.

To achieve Goal G3 (*"Analyze the FRED robot with the purpose of evaluating its characteristics related to the Animacy scale, based on user perception"*) participants evaluated FRED according to the Animacy dimension that was associated with the AN metric. Goal G3 was achieved and participants, as with the AN metric, were neutral regarding the Animacy dimension. Gender and professional background did not influence participants' responses.

To achieve Goal G4 (*"Analyze the FRED robot with the purpose of evaluating its characteristics related to the Likeability scale, based on user perception"*), participants evaluated FRED according to the Likeability dimension that was associated with the LK metric. Goal G4 was achieved and participants evaluated FRED with the highest possible value ($Md = 5$). Although professional background did not influence participants' perception, gender did. Females had a better perception of the robot, in relation to the Likeability dimension, than males.

To achieve Goal G5 (*"Analyze the FRED robot with the purpose of evaluating its characteristics related to the Perceived Intelligence scale, based on user perception"*) participants evaluated FRED according to the Perceived Intelligence dimension that was associated with the PI metric. Goal G5 was achieved and participants evaluated FRED as partially competent, partially knowledgeable, partially responsible, partially intelligent and partially sensitive. Gender and professional background did not influence participants' perceptions of FRED in relation to the Perceived Intelligence dimension.

To achieve Goal G6 (*"Analyze the FRED robot with the purpose of evaluating whether, based on the user's perception, children would like to interact with FRED robot"*). For this Goal the Perceived Child Acceptance (PCA) metric was defined. Goal G6 was achieved and participants strongly agreed that *"Children would like to interact with FRED robot"*. Gender and professional background did not influence participants' perceptions regarding the PCA metric.

In addition to the analyses and discussions based on the metrics defined for each question of each Goal, an analysis was also carried out with the aim of exploring the semantic items of the scales of each of the dimensions. This analysis resulted in several insights that can contribute to the improvement of FRED in each of the dimensions evaluated.

6.4 Final Remarks

This chapter introduced FRED, the low-cost affective social robotics platform proposed in this work. This platform represents another contribution of this thesis. Its 3D components, hardware and software architecture, and functionalities were presented. FRED served as the fourth instantiation case for the framework, specifically regarding the use of the Robot Abstraction Layer in building its control system.

Furthermore, this chapter presented the methodology and evaluation of the FRED

robot's affective capabilities and characteristics based on the dimensions outlined in the Godspeed Questionnaire Series (GQS). To this end, the GQM paradigm was employed, defining six objectives accompanied by their respective questions and metrics. The evaluation was based on videos, and 113 participants completed online questionnaires. By the end of the evaluation, all six objectives had been achieved. Together, the implementation and evaluation of FRED provide evidence regarding Research Question 1 by demonstrating a concrete approach to expressing affective states through multiple non-verbal communication elements and by investigating how these expressions are recognized by users.

The next chapter presents an evaluation of the extended versions of the language and simulator (EvaML 2.0 and EvaSIM 2.0) proposed in this thesis, investigating their usability and user experience within a learning and development context. It also presents a qualitative assessment of the experience of the HRI researchers responsible for the instantiations of the GRACE robot, presented in Section 4.3, and the FRED MX robot, presented in Section 4.4.

7 Evaluation of EvaML 2.0, EvaSIM 2.0 and Framework Applicability

As presented in Section 5, two other contributions of this doctoral thesis are the extensions to the EvaML language and the EvaSIM simulator. These tools support the programming, testing, simulation, and execution of interaction scripts for the EVA robot. Their original versions were extended to support the new capabilities of the EVA 2.0 robot, presented in Section 4.2, as well as the new functionalities provided by the framework’s affective component.

This chapter presents an evaluation of EvaML 2.0 and EvaSIM 2.0, investigating usability and user experience regarding these new versions within a learning and development context. The evaluation was conducted through a study involving hackathon participants with little to no prior programming experience, enabling an analysis of the tools’ usability and ease of use.

Furthermore, this chapter presents an exploratory evaluation based on case studies regarding the instantiations described in Sections 4.3 and 4.4 using the GRACE and FRED MX robots. A qualitative analysis is presented via a survey consisting of 16 questions. The aim of this survey is to investigate the use of the framework in real-world HRI projects conducted by researchers from other institutions.

This chapter is organized as follows. Section 7.1 presents the evaluation of EvaML 2.0 and EvaSIM 2.0. Section 7.2 presents two real-world use cases of the framework application.

7.1 Evaluation of EvaML 2.0 and EvaSIM 2.0

This section evaluates the user experience with EvaML 2.0 and the usability of EvaSIM 2.0, which result from the extension of the original versions of EvaML and EvaSIM presented in Appendix A. This evaluation took place in an educational context involving

participants with limited or no prior programming experience. More specifically, this study aims to investigate how these tools assist novice users in the process of implementing and executing applications for a social robot, considering both the use of the programming elements provided by the language and the simulator's effectiveness as an environment for simulation and script execution.

7.1.1 Study Design

The study was conducted within the context of an educational hackathon aimed at introducing participants with little to no prior programming experience to application development for social robots. The activity was organized in collaboration with an initiative encouraging girls to pursue careers in STEM (Science, Technology, Engineering, and Mathematics), involving elementary and high school students as well as teachers from public schools. Two editions of the hackathon were held. The first edition, conducted in 2024, took a purely exploratory approach, without formal evaluation. The second edition, held in 2025 with a different group of participants, adopted a structured approach to evaluate the user experience and the usability of the programming language and the simulator, respectively.

The following sections describe both editions of the hackathon, moving from an exploratory study in 2024 to a structured and evaluated experiment in 2025.

7.1.1.1 First Hackathon Edition (2024)

The first educational hackathon followed a structured sequence of guided sessions, combining introductory presentations, a robot demonstration, and collaborative design and development tasks. The activity involved a group of participants with limited or no prior programming experience and aimed to explore the use of social robotics tools in an introductory learning context (SOUZA et al., 2025).

The group consisted of nine students (from elementary and high school) and three teachers, all female. Six of the students and the three teachers had participated in a four-hour introductory course on Python programming about twelve months earlier, whereas the other three students had no prior experience with programming concepts. In this context, the activity was designed as an exploratory study, prioritizing hands-on interaction and collaborative learning over formal assessment. The activity was organized into six weekly 90-minute sessions, allowing participants to progressively engage with the robot's

programming environment, programming language, and simulator.

Initially, participants were introduced to the robot's capabilities, including its verbal and non-verbal communication, through a demonstration using the physical robot. Subsequently, they were guided through the ideation process using design thinking principles. As part of this process, participants were organized into three teams of four members, each comprising three students and one teacher, and tasked with designing applications that leveraged the robot's functionalities.

The subsequent sessions focused on hands-on activities involving the EvaML 2.0 programming language and the EvaSIM 2.0 simulator. Initially, participants individually explored the basic concepts and functionalities of these tools and later collaborated within their teams to implement the proposed solutions. Throughout the process, participants were supported by facilitators, undergraduate Computer Science students with prior experience in EvaML 2.0, who assisted with technical issues and guided the programming activities. The activity concluded with the presentation of the developed applications, which were first tested in the EvaSIM 2.0 simulator and subsequently executed on the physical EVA robot.

Observations made during the activity, together with an analysis of the EvaML scripts produced by the teams, provided preliminary insights into how participants interacted with the language and the simulator. These initial insights served as the basis for designing a formal evaluation for the second edition of the hackathon, which adopted a structured methodological approach to assess user experience and usability.

7.1.1.2 Second Hackathon Edition (2025)

The second edition of the hackathon followed a structure similar to the first, maintaining the same sequence of guided sessions. The activity was organized as a series of weekly meetings that combined introductory presentations with collaborative development tasks. Initially, participants were introduced to the robot's capabilities through an in-person demonstration and were subsequently guided through the ideation process using design thinking principles. As in the first edition, they were organized into small teams and tasked with designing applications that leveraged the robot's functionalities.

In contrast to the 2024 edition, the 2025 hackathon was designed as a structured experiment involving a distinct group of participants, followed by an assessment of user experience and usability. Subsequent sessions focused on hands-on activities in which



Figure 63: Participants performing hands-on activities during the educational hackathon and testing scripts developed in EvaML 2.0 on the robot using the EvaSIM 2.0 simulator.

participants worked with EvaML 2.0 and the EvaSIM 2.0 simulator, initially exploring the tools individually and later collaborating in teams to develop the solutions proposed during the initial phase. Throughout the process, as in the 2024 edition, participants were supported by undergraduate students with prior experience on the platform, who assisted with technical issues and guided the programming activities. Just as in the first edition, the event concluded with a presentation of the applications developed and executed on the physical robot. Figure 63 shows images from the 2025 Hackathon, including photos of participants working with the programming language and the simulator, and testing their scripts on the physical robot, EVA.

7.1.2 Methodology

The evaluation of EvaML 2.0 and EvaSIM 2.0 is structured according to the Goal Question Metric (GQM) paradigm, which provides a systematic framework for aligning the objectives of the study with the research questions and the corresponding metrics used for data collection and analysis. Following this approach, two main goals are defined, each addressing a specific tool.

The defined goals are presented in Table 32, which summarizes the focus of the evaluation for both EvaML 2.0 and EvaSIM 2.0. Based on these goals, a set of research questions is formulated to guide the empirical investigation. The research questions associated with Goal G1, related to the evaluation of user experience with EvaML 2.0, are presented in

Table 32: GQM goals defined for the evaluation of EvaML 2.0 and EvaSIM 2.0.

Goal	Description
G1	Analyze EvaML 2.0, with the purpose of evaluating its user experience from the users' point of view.
G2	Analyze EvaSIM 2.0, with the purpose of evaluating its usability from the users' point of view.

Table 33: Questions for Goal G1.

Question	Description
Q1	How do participants perceive the user experience of EvaML 2.0?
Q2	Does the perceived user experience of EvaML 2.0 improve after hands-on interaction with the language?

Table 33, while the research question for G2, concerning the usability of EvaSIM 2.0, are shown in Table 34.

To ensure that each research question is supported by measurable evidence, a corresponding set of metrics is defined. The metrics associated with G1 are presented in Table 35 and are based on the User Experience Questionnaire – Short Version (UEQ-S), capturing both pragmatic and hedonic dimensions of user experience, as well as their evolution after hands-on interaction. The metric for G2, presented in Table 36, is based on the System Usability Scale (SUS), providing a standardized measure of perceived usability and its interpretation according to established benchmarks. Together, these tables provide a structured representation of the evaluation design, explicitly linking each goal to its corresponding questions and metrics.

7.1.3 Experiment

This study aims to evaluate the user experience with the EvaML 2.0 programming language and the usability of the EvaSIM 2.0 simulation environment in an educational context. To this end, a study with a quantitative approach was conducted, structured as a short-term longitudinal experiment. The study took place over three days in an intensive manner and was organized in a hackathon format.

The evaluation was conducted at two distinct times: (i) immediately after the intro-

Table 34: Question for Goal G2.

Question	Description
Q3	How do participants perceive the usability of EvaSIM 2.0?

Table 35: Metrics for G1 questions.

Metric	Description	Question
UEQ-S Scores	Mean values of Pragmatic and Hedonic Quality dimensions measured on Day 1 and Day 3.	Q1
UEQ-S Score Change	Difference between UEQ-S Pragmatic and Hedonic Quality scores measured on Day 1 and Day 3, used to assess changes in participants' perception after interaction with EvaML 2.0. The statistical significance of these differences will be evaluated using the Wilcoxon signed-rank test.	Q2

Table 36: Metric for G2 question.

Metric	Description	Question
Mean SUS Score	Mean SUS score calculated across all participants. The resulting value is interpreted based on established usability benchmarks for the System Usability Scale (SUS).	Q3

duction to the EvaML 2.0 language (Day 1), when the UEQ-S was administered to assess the user experience specifically related to EvaML 2.0; and (ii) after the practical use of EvaML 2.0 and EvaSIM 2.0 in developing and executing scripts on a physical robot (Day 3), when the UEQ-S was administered again to assess changes in user experience with EvaML 2.0, and the SUS was used to evaluate the usability of the EvaSIM 2.0 simulator. This approach allows for an analysis of how user experience evolves over time and following practical interaction with both tools.

7.1.3.1 Participants and Recruitment

The experiment involved 14 participants: 11 female students from two public elementary and high schools and three female teachers. All participants were female. The students had a mean age of 16 years ($SD = 1.4$), while the teachers were over 40 years old.

Participants were organized into three groups: Group I (STARS), consisting of three students and a teacher; Group II (DIVAS ALTIVAS), also consisting of three students and a teacher; and Group III (ÂMINA), consisting of five students and a teacher. Regarding prior experience, five participants reported having previous programming experience. Of these five, four indicated limited experience, having programmed only a few times, with one belonging to the DIVAS group and three to the ÂMINA group. The fifth participant, also a member of the Âmina group, reported programming more regularly. Furthermore,

six students reported having previously interacted with the physical robot: four from the DIVAS group and two from the ÂMINA group.

Participants were recruited through outreach activities conducted at the participating schools. Teachers involved in the project visited the schools to present introductory sessions covering topics such as technology, programming, and the EVA robot, followed by live demonstrations of the robot's capabilities.

7.1.3.2 Instruments and Data Collection

The user experience with EvaML 2.0 was evaluated using the UEQ-S, which consists of pairs of opposing adjectives arranged on a seven-point Likert scale. The instrument captures two main dimensions: Pragmatic Quality, associated with usability, efficiency, and ease of use; and Hedonic Quality, related to the extent to which the system is perceived as stimulating, pleasant, and attractive. Using the UEQ-S in this study allows for capturing both the functional and emotional aspects of the participants' interaction with the EvaML 2.0 language.

In addition, the usability of EvaSIM 2.0 was evaluated using the SUS, a widely adopted standardized questionnaire for assessing perceived usability. SUS consists of ten items formulated as statements, with responses collected on a five-point Likert scale ranging from "Strongly disagree" to "Strongly agree." In this study, the original SUS items were adapted to explicitly refer to the EvaSIM 2.0 simulator, ensuring that participants' responses were directly associated with their interaction with the system. The SUS provides a global measure of perceived usability, allowing for the assessment of how effectively and easily participants can interact with the simulator during the development and execution of robot applications.

7.1.3.3 Experimental Procedure

The experiment was conducted over three consecutive days in a laboratory setting, following an intensive and structured format. The procedure was designed to provide participants with a progressive learning experience, combining introductory sessions, guided practice, and collaborative development tasks involving EvaML 2.0 and EvaSIM 2.0. On the first day, participants attended an introductory session presenting the key capabilities of the EVA robot, followed by an introduction to the programming language. After this initial exposure, participants completed the UEQ-S to record their first impressions of

EvaML 2.0. On the second day, participants engaged in practical programming activities using the language and the simulator, working initially individually and later in small groups to implement their applications, with support from instructors experienced with the platform. On the third day, participants finalized their solutions and executed them on the physical EVA robot, completing the transition from simulation to real-world implementation. At the end of the activity, the UEQ-S was administered again to assess changes in the user experience with EvaML 2.0, and the SUS was used to evaluate the usability of the EvaSIM 2.0 simulator.

7.1.3.4 Data Analysis

The data analysis focused on two complementary aspects: the user experience with EvaML 2.0 and the usability of EvaSIM 2.0. UEQ-S responses collected on Day 1 and Day 3 were analyzed to assess changes in user experience over time, considering the dimensions of Pragmatic Quality and Hedonic Quality. Descriptive statistics were calculated for each dimension, and a Wilcoxon signed-rank test was performed to compare the two measurement points, enabling the identification of statistically significant differences in participants' perceptions following hands-on interaction with the language.

For the evaluation of EvaSIM 2.0, SUS scores were calculated using standard procedures, resulting in an individual usability score for each participant. These scores were then aggregated to obtain an overall measure of perceived usability. The mean SUS score was interpreted based on established benchmarks, with values of 68 or higher indicating acceptable usability.

Taken together, these analyses provide a comprehensive assessment of how participants experienced the EvaML 2.0 programming language and perceived the usability of the EvaSIM 2.0 simulator, thereby grounding the evaluation of these new, extended versions of the tools.

7.1.4 Results

This section presents the results of the evaluation conducted according to the GQM paradigm. The evaluation was carried out within the context of the educational hackathon described in Section 7.1.1, in which participants engaged in guided practical activities involving the development of applications for a social robot, utilizing the EvaML 2.0 and the EvaSIM 2.0.

Table 37: Descriptive statistics and Wilcoxon signed-rank test results for UEQ-S dimensions (Pragmatic and Hedonic) across Day 1 and Day 3.

Dimension	Day 1		Day 3		Statistic		
	Mean	SD	Mean	SD	W	p	Effect Size (r)
Pragmatic Quality	1.41	0.87	1.52	0.80	45.0	0.665	0.154
Hedonic Quality	2.20	0.75	2.52	0.54	37.5	0.082	0.667

First, the user experience with EvaML 2.0 is examined using the UEQ-S instrument, considering pragmatic and hedonic quality dimensions and their evolution following practical interaction, in order to assess changes in perceived user experience. Next, the usability of EvaSIM 2.0 is evaluated using the SUS, providing descriptive statistics and standard interpretability measures to characterize participants' perceptions of the simulator. Together, these results offer empirical evidence to support the evaluation of the proposed versions of the robot programming language and simulator from the perspectives of both user experience and usability.

7.1.4.1 G1 - User Experience with EvaML 2.0

The user experience with EvaML 2.0 was evaluated by analyzing the experience perceived both before and after hands-on interaction with the programming language. To address G1, values for the Pragmatic Quality and Hedonic Quality dimensions were calculated based on the application of the UEQ-S.

Table 37 presents the UEQ-S results obtained after the initial exposure to EvaML 2.0 (Day 1). Figure 64 shows the mean scores for Pragmatic Quality ($Mean = 1.41$; $SD = 0.87$), Hedonic Quality ($Mean = 2.20$; $SD = 0.75$), and Overall ($Mean = 1.80$; $SD = 0.64$), plotted against the UEQ-S reference scale.

After hands-on interaction (Day 3), the mean scores were calculated for Pragmatic Quality ($Mean = 1.52$, $SD = 0.80$), Hedonic Quality ($Mean = 2.52$, $SD = 0.54$), and Overall ($Mean = 2.02$, $SD = 0.55$). The results are also presented in Table 37. Figure 65, presents this values positioned within the UEQ-S benchmark scale.

Table 37 presents the results of the Wilcoxon test for both dimensions in the comparison between the UEQ-S values for the two days. The results for Pragmatic Quality were $W = 45.0$, $p = 0.665$, and $r = 0.154$ (effect size) while for Hedonic Quality, $W = 37.5$, $p = 0.082$, and $r = 0.667$. The p values indicate that no statistically significant differences

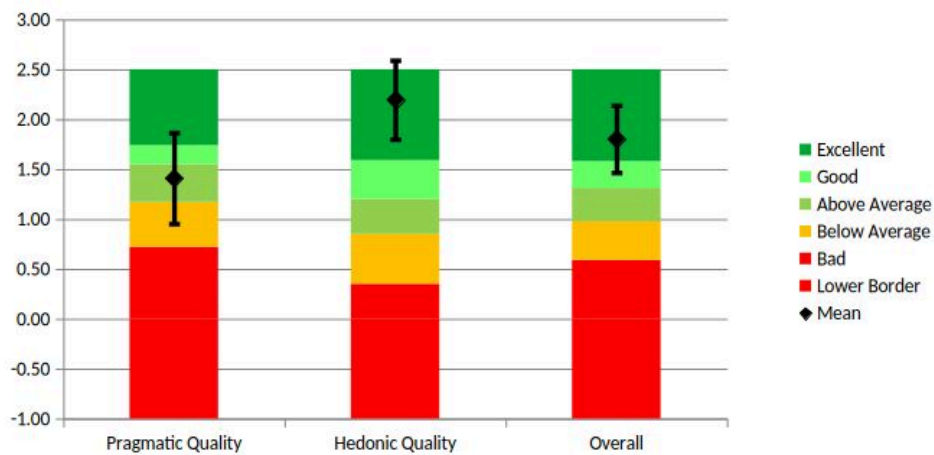


Figure 64: UEQ-S results for EvaML 2.0 after initial exposure (Day 1). Mean scores for Pragmatic Quality, Hedonic Quality, and Overall user experience.

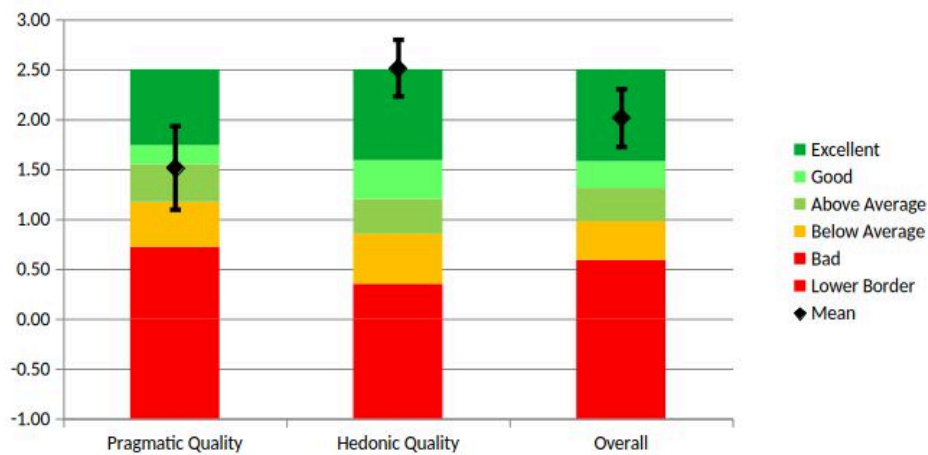


Figure 65: UEQ-S results for EvaML 2.0 after hands-on interaction (Day 3). Mean scores for Pragmatic Quality, Hedonic Quality, and Overall user experience.

were observed in either case.

7.1.4.2 G2 - Usability of EvaSIM 2.0

The usability of EvaSIM 2.0 was evaluated according to the GQM paradigm, addressing G2. This analysis is guided by the corresponding research question regarding whether the system provides an adequate level of usability, and is achieved through metrics derived from the System Usability Scale (SUS), including the overall SUS score and standard interpretability measures.

Table 38 summarizes the descriptive statistics of the SUS scores obtained from the participants. The mean SUS score was 66.25 ($SD = 13.3$). To support the interpretation of this result, Figure 66 presents the standard acceptability ranges and adjective ratings

Table 38: Descriptive statistics and interpretation of System Usability Scale (SUS) scores for EvaSIM 2.0, including acceptability range and adjective rating.

Mean SUS Score	SD	M_d	IQR	Min	Max	Acceptability Range	Adjective Rating
66.25	13.3	65	11.3	42.5	85	Marginally acceptable	OK

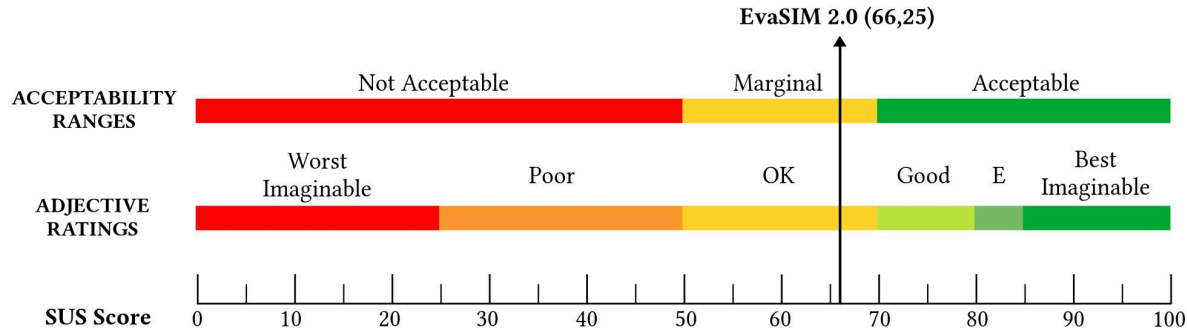


Figure 66: Standard interpretation of SUS scores based on acceptability ranges and adjective ratings, adapted from (BANGOR; KORTUM; MILLER, 2008).

associated with SUS scores. This visualization provides a reference for understanding the overall usability level indicated by the participants’ responses.

Table 39 presents the descriptive statistics of participants’ responses for each SUS item, including median (M_d) and interquartile range (IQR). Figure 67 shows the distribution of responses for each item using boxplots, including the interquartile ranges and potential outliers across different SUS items.

7.1.5 Discussion

This section discusses the results obtained in the evaluation of the EvaML 2.0 language and the EvaSIM 2.0 simulator, aiming to assess user experience and usability from the user’s perspective. In general, the findings suggest that both EvaML 2.0 and EvaSIM 2.0 were perceived positively. The variability observed in specific SUS items highlights areas where improvements in EvaSIM 2.0 can enhance user interaction. Furthermore, the analysis of the change in the UEQ-S score between Day 1 and Day 3 provides evidence of how hands-on interaction influenced users’ perceptions of the language over time. These results are examined based on the defined objectives, integrating both quantitative and qualitative findings.

Table 39: Descriptive statistics (median and interquartile range, IQR) of participant responses for each System Usability Scale (SUS) item.

Item	Statement	M_d	IQR
1	I think that I would like to use the EvaSIM 2.0 simulator frequently.	5	1
2	I found the EvaSIM 2.0 simulator unnecessarily complex.	1	1
3	I thought the EvaSIM 2.0 simulator was easy to use.	3.5	1
4	I think that I would need the support of a technical person to be able to use the EvaSIM 2.0 simulator.	4	1
5	I found the various functions of the EvaSIM 2.0 simulator were well integrated.	5	1
6	I thought there was too much inconsistency in the EvaSIM 2.0 simulator.	2	2
7	I would imagine that most people would learn to use this system very quickly.	4	1
8	I found the EvaSIM 2.0 simulator very cumbersome to use.	2.5	1
9	I felt very confident using the EvaSIM 2.0 simulator.	4	0.75
10	I needed to learn a lot of things before I could get going with the EvaSIM 2.0 simulator.	4	1.75

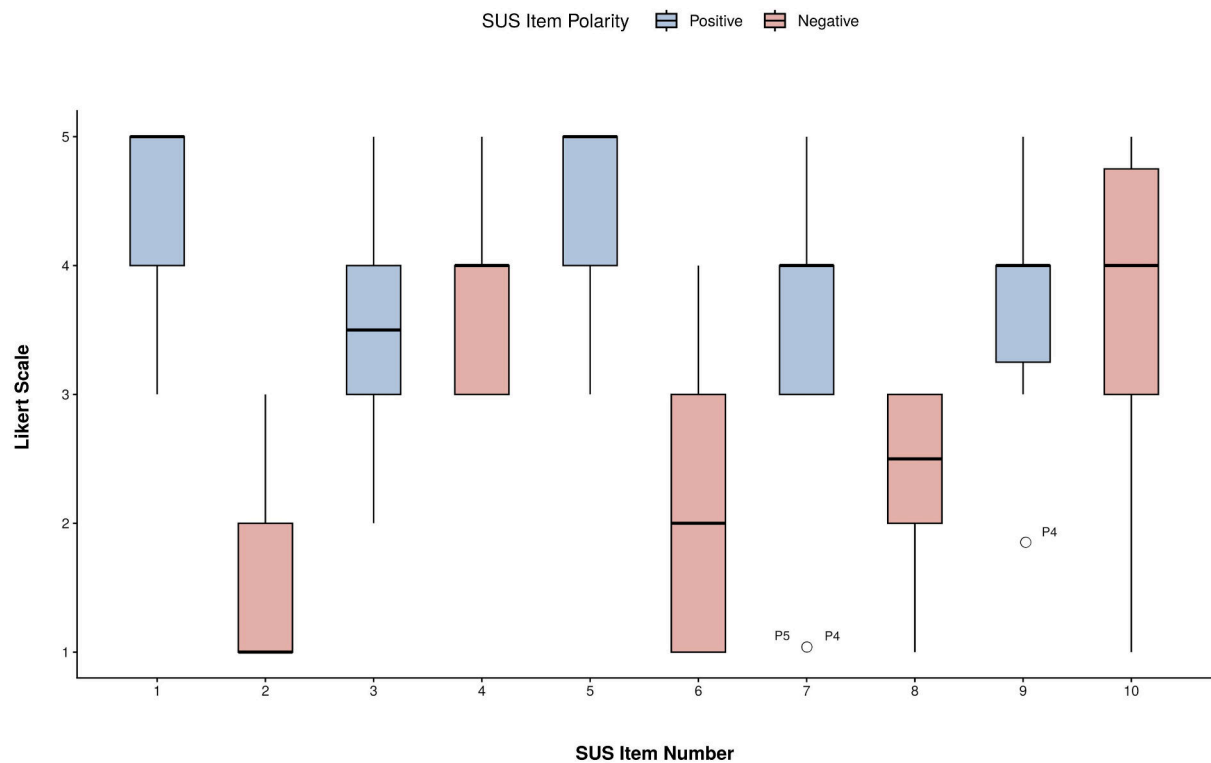


Figure 67: Distribution of participant responses for each System Usability Scale (SUS) item. Median and interquartile range, with items differentiated by polarity (positive and negative).

7.1.5.1 User Experience with EvaML 2.0

The results for EvaML 2.0 suggest a generally positive perception of user experience among novice users. The results of the user experience evaluation for the EvaML 2.0 language were positive from Day 1, the students' first contact with the language. As shown in Figure 64, the value obtained for the Pragmatic Quality dimension shows that, on the first day, the language was evaluated as "Above Average" and for Hedonic Quality, it was classified as "Excellent". The Overall UEQ-S score for EvaML 2.0 was classified as "Excellent".

For Day 3, the values of the dimensions increased. A very subtle increase occurred in the Pragmatic Quality dimension (0.11), bringing it very close to being classified as "Good" in Figure 65. A more significant increase occurred in the Hedonic Quality dimension (1.0), maintaining the "Excellent" classification. There was an overall increase of 0.22, maintaining the Overall UEQ-S score rating as "Excellent". The higher mean values observed for Hedonic Quality compared to Pragmatic Quality suggest that participants perceived the interaction with the language as attractive, interesting, and stimulating. However, the comparatively lower values for Pragmatic Quality suggest that certain aspects related to usability, such as clarity, efficiency, or ease of use, may require further refinement.

Despite the increases in the mean values for each dimension, especially for Hedonic Quality, the statistical test did not show a significant difference in the comparison between the UEQ-S scores of Day 1 and Day 3, indicating that there was no increase after the hands-on interaction. Although the statistical test did not reveal a significant difference for both dimensions, for the Hedonic Quality dimension ($p = 0.08$), the observed effect size ($r = 0.667$) suggests a relevant change in user perception. This result may suggest that interaction with the language may have influenced participants' experience, even if this effect was not statistically significant within the sample.

These results suggest that EvaML 2.0 provides a positive and engaging initial experience for novice users, particularly in terms of hedonic aspects of interaction. While the observed improvements between Day 1 and Day 3 were not statistically significant, the increase in mean values and the large effect size for Hedonic Quality suggest a tendency toward improved user perception after hands-on interaction. At the same time, the comparatively lower values for Pragmatic Quality highlight opportunities to improve functional aspects related to usability, such as clarity and efficiency.

7.1.5.2 Usability of EvaSIM 2.0

The usability of EvaSIM 2.0 was evaluated using the System Usability Scale (SUS) from the user's point of view. The interpretation of this value was done using the standard interpretation of SUS scores based on acceptability ranges and adjective ratings, presented in Figure 66.

The mean SUS score for EvaSIM 2.0 indicate that the system is, on an acceptability scale, within the range considered "Marginally Acceptable" (66.25), a value close to the mean value of 68.00. On an adjective scale, it is classified as "OK".

Table 39 presents the median and interquartile range values for the users' responses (on a Likert scale) for each of the SUS items. Figure 67 presents the same values through a boxplot graph. Analysis of responses at the item level can provide additional information about users' perceptions. These results provide complementary insights and do not replace the interpretation based on the mean SUS score.

For item 3, which is a positively polarized item, users demonstrated neutrality ($Md = 3.5$, $IQR = 1$) regarding the statement that the simulator was easy to use. For item 4, which has negative polarity, the median ($Md = 4$) and interquartile range ($IQR = 1$) values suggest that users consistently agreed that they would need technical support to use EvaSIM 2.0. For item 10, the median value is 4 ($Md = 4$), indicating that users partially agree that they would need to learn many things before using the simulator. However, the relatively large interquartile range ($IQR = 1.75$) reflects a high level of variability in the responses. For item 6, despite the median value ($Md = 2$), showing that users partially disagree that there was much inconsistency in the simulator, a relatively large interquartile range ($IQR = 2$) indicates considerable variability in users' perceptions. A plausible explanation for this behavior may be the diversity of the participant group, which included individuals with different levels of experience and familiarity with technology.

The mean SUS score indicates that EvaSIM 2.0 is in the "Marginally Acceptable" range on the acceptability scale and is classified as "OK" on the adjective rating scale. Although this result suggests that novice users were able to interact with the system, item-level analysis reveals important details, particularly regarding ease of learning and the need for technical support. The variability observed among the items, including higher variability in some responses, suggests that the system's usability is not perceived uniformly among participants. These findings indicate that, while the system is generally usable, improvements related to clarity, guidance, and reduction of perceived complexity

could contribute to a better and more accessible user experience.

7.1.6 Conclusion

This section presented the evaluation of the EvaML 2.0 programming language and the EvaSIM 2.0 simulator. The results suggest that EvaML 2.0 provides a positive user experience, as reflected by consistently favorable scores in both Pragmatic and Hedonic Quality dimensions. Although no statistically significant improvement was observed after hands-on interaction, participants were able to quickly understand and effectively use the language within a limited interaction period, highlighting its potential as an accessible programming tool for educational and exploratory contexts.

Regarding EvaSIM 2.0, the results show that the simulator achieves a "Marginally Acceptable" level of usability, with an average SUS score corresponding to an "OK" rating on an adjective scale. Although the system was generally perceived as usable, the variability observed in participants' responses suggests that usability is not perceived homogeneously by users.

These findings provide evidence that the proposed extended versions of the tools support the development of applications for social robots in educational environments.

The evaluation was conducted with a relatively small sample in a hackathon setting, which may limit both the statistical power of the analyses and the external validity of the findings. In particular, The small sample size, combined with variability in participants' responses, may have made it more difficult to detect statistically significant differences. Consequently, the results should be interpreted with caution and viewed as indicative rather than conclusive.

7.2 Framework Assessment in HRI Research Contexts

Chapter 4, specifically Sections 4.3 and 4.4, presented two instantiations conducted by HRI researchers at other institutions, one in Switzerland and the other in Mexico. The robots introduced were GRACE (based on the EVA robot) and FRED MX (based on the FRED robot). This section presents an exploratory evaluation using a 16-question survey. The survey aims to investigate the use of the framework in real-world HRI projects conducted by researchers in the field. The questions seek to determine whether they can effectively use the framework to design, implement, adapt, and deploy social robot

applications tailored to their specific research objectives. In this context, the successful implementation of projects based on the framework can provide evidence of its applicability and its potential to support research and researchers in the field of social robotics.

7.2.1 Methodology

7.2.1.1 Evaluation Design

This evaluation was conducted as an exploratory study, focusing on feedback from users who adopted the framework in independent research projects. Data were collected via a questionnaire containing both closed- and open-ended questions, covering topics such as the learning curve, required support, framework extensibility, perceived benefits, difficulties encountered, and recommendations for future improvements. The study investigated how independent researchers employed the framework in real-world HRI projects.

The evaluation involved three researchers who applied the framework to two distinct HRI projects conducted in Switzerland and Mexico. These projects, based on versions derived from the EVA and FRED robots, offered complementary perspectives on the use of the framework, robot assembly activities, integration, and customization.

Given the limited number of participants and the exploratory nature of the study, the findings are not intended to be generalized to a broader population of users. Instead, the results provide evidence that the framework can support various types of HRI research projects, involving both the reuse of existing components and the development of custom solutions.

7.2.1.2 Research Questions

To guide the evaluation and provide a structured analysis of the researchers' experiences, a set of research questions was defined. These questions focus on three complementary aspects of using the framework: the researchers' experiences in employing it in their projects, the adaptations and extensions made to meet specific project requirements, and the benefits and challenges encountered during its use.

The following research questions were defined:

- **RQ1:** How did researchers experience the use of the framework in their projects?
- **RQ2:** How was the framework adapted or extended to meet project requirements?

- **RQ3:** What strengths and challenges were encountered during framework use?

RQ1 focuses on understanding how researchers experienced using the framework in their projects, aiming to gain insight into aspects such as the perceived learning curve, the level of assistance required, and the researchers' confidence in supporting others interested in using the framework.

RQ2 examines how the framework was employed, adapted, or extended to meet the requirements of different research projects. This includes configuring existing functionalities, adapting available components, integrating new capabilities, and developing complementary tools.

RQ3 seeks to identify the benefits and challenges encountered while using the framework, focusing on recurring issues related to development effort, the reuse of existing components, documentation, system configuration, and hardware or software integration.

7.2.1.3 Data Collection

Data were collected via a questionnaire combining rating-scale items and open-ended questions, administered to researchers who had used the framework in actual HRI research projects. The questionnaire was designed to gather both quantitative and qualitative information. Closed-ended questions utilized five-point rating scales, addressing aspects such as the perceived learning curve, the level of assistance required to use the framework, and the difficulty of extending or modifying its functionalities. Open-ended questions invited participants to describe their projects, report difficulties encountered during development, explain how the framework facilitated their work, identify desired improvements, and provide general feedback on the experience. The complete questionnaire and the participants' responses are provided in [Appendix C](#).

The questionnaire was completed by three researchers involved in two independent HRI projects: the GRACE project, conducted at ETH Zurich, and the FRED MX project, conducted at CICESE. Together, these responses provided complementary perspectives on the use of the framework in different research and development contexts.

The goal of the questionnaire was not to formally evaluate usability or user experience using standardized instruments, but rather to document the experiences of researchers who adopted the framework in practice.

7.2.1.4 Data Analysis

Given the exploratory nature of the evaluation and the small number of participants, the collected data were analyzed primarily through descriptive and qualitative approaches.

Responses to closed-ended questions were analyzed descriptively at the individual level. They served to characterize each participant's experience regarding aspects such as the perceived learning curve, the level of assistance required, and the difficulty of extending or modifying the framework's functionalities. Subsequently, comparative summaries were used to identify similarities and differences among the reported experiences. Responses to open-ended questions were reviewed and examined to identify recurring observations regarding the use of the framework. Similar observations were grouped into broader categories representing benefits, challenges, and common usage patterns reported across the different cases.

Combining individual descriptive analysis with the review of open-ended responses provided a comprehensive perspective on the use of the framework in the analyzed projects, including how it was employed, adapted, and perceived within different research contexts.

7.2.2 Participants and Research Project Scenarios

The framework was adopted in two independent HRI research projects conducted at institutions outside the original development environment. These projects involved distinct research objectives, target populations, and technical requirements, providing an opportunity to examine the framework's applicability from the perspective of researchers in different contexts.

7.2.2.1 GRACE Robot

The first research project considered in this evaluation involved the development of the GRACE robot at ETH Zurich. The project was conducted by a master's student (P1) working in areas related to interaction design, social robotics, and cognitive interventions. GRACE was developed as a derivative of the EVA robot and used in a research project focused on cognitive interventions delivered by a social assistive robot. Four activities were implemented, two of which were based on Cognitive Stimulation Therapy. The application combined rule-based interaction scripts with AI-generated dialogue, requiring the integration of ChatGPT into the system architecture. The robot was initially evaluated

in a controlled study with healthy adults and subsequently used in a study with healthy older adults, with the long-term goal of supporting future research involving people with dementia.

7.2.2.2 FRED MX Robot

The second project involved the FRED MX robot, developed in Mexico. A master's student (P2) at CICESE, employed the framework to investigate affective conversational strategies aimed at supporting procrastination management in academic settings. The project focused primarily on interaction design and the evaluation of socio-affective behaviors. Unlike the GRACE project, no major extensions to the framework were required. Instead, existing conversational and affective behaviors were configured and adapted to support the study objectives.

7.2.2.3 Technical Development of FRED MX

A complementary perspective was presented by an undergraduate student (P3) who participated in the same FRED MX project described in the previous section. While the master's student focused primarily on the design and evaluation of socio-affective interactions, the undergraduate student was responsible for technical development activities related to the robot's construction and operation. These activities included assembling the platform, troubleshooting communication issues between the ESP32 controller and the host computer, and developing a C# control panel to facilitate the robot's operation. This contribution offers additional insights into the use of the framework from a technical and engineering perspective, complementing the research-oriented experience reported in the FRED MX project.

Together, the participants (P1, P2 and P3) provide complementary perspectives on framework use, encompassing both research-oriented and technical development activities. Their projects involved different application domains, levels of customization, and technical requirements, providing a diverse view of the framework's applicability, and extensibility in real-world HRI scenarios.

7.2.3 Results

The following subsections present the results according to the three research questions defined in Section 7.2.1.2. Section 7.2.3.1 addresses research question 1 (*"RQ1: How*

Table 40: Summary of responses regarding framework experiences reported by participant P1 (GRACE project) and participants P2 and P3 (FRED MX project).

Aspect	GRACE (P1)	FRED MX (P2)	FRED MX (P3)
Learning Curve (1 = Very difficult, 5 = Very easy)	5	3	4
Required Help (1 = No help, 5 = Excessive help)	3	3	3
Framework Extension Required	Yes	No	Yes
Extension Difficulty (1 = Very difficult, 5 = Very easy),	2	N/A	4
Comfortable Guiding Others	Partially	Yes	Yes

did researchers experience the use of the framework in their projects?"), Section 7.2.3.2 addresses research question 2 ("*RQ2: How was the framework adapted or extended to meet project requirements?*"), and Section 7.2.3.3 addresses research question 3 ("*RQ3: What strengths and challenges were encountered during framework use?*").

7.2.3.1 Researchers' Experiences with the Framework

Table 40 summarizes the responses regarding the use of the framework in the GRACE and FRED MX projects. The reported learning curve ranged from neutral to very easy. Researcher P1 gave the highest rating (5/5), while researcher P2 gave a neutral rating (3/5). The technical developer (P3) reported an easy learning experience (4/5). All participants indicated needing a moderate level of assistance when working with the framework (3/5). Regarding the ability to support other researchers interested in using the framework, two participants (P2 and P3) reported feeling comfortable offering guidance, whereas one participant (P1) reported feeling only partially comfortable doing so.

7.2.3.2 Framework Extensibility

The framework was employed in various ways across the analyzed projects. In the GRACE project, the researcher integrated ChatGPT, modified existing voice and facial expression functionalities, and incorporated new voice-capture hardware. In the FRED MX research project, existing conversational and affective behaviors were adapted to meet the study's objectives. Additionally, an extra control panel was developed in C# to facilitate robot operation. These modifications illustrate different ways of utilizing the framework, ranging from behavior configuration to the integration of new functionalities and support tools.

7.2.3.3 Framework Strengths and Challenges

To complement the information obtained through the closed-ended questions, responses to the open-ended questions were analyzed to identify recurring observations regarding the use of the framework. These observations were organized into categories representing strengths and challenges frequently reported during the framework's use.

Reduction of Development Effort. Researchers reported that the framework reduced the development effort required to implement their projects. Researcher P1 stated that the availability of hardware and software components enabled the implementation of a social robot platform as part of a master's thesis. Similarly, researcher P2 reported that the framework reduced the need for low-level implementation, allowing for a greater focus on interaction design and evaluation activities.

Reuse of Existing Components. The responses indicate that existing framework components were reused in both projects. In the GRACE project, hardware and software infrastructure previously developed to support the implementation of cognitive intervention activities was utilized. In the FRED MX project, existing conversational and affective behaviors were adapted to the study's objectives. These responses illustrate the use of previously available components in different application scenarios.

Framework Documentation and Understanding. All three participants reported difficulties regarding the documentation and understanding of the framework. Researcher P1 mentioned challenges related to grasping the code structure, identifying component responsibilities, and the limited availability of documentation in English. Similarly, researcher P2 reported difficulties in understanding how the framework's modules and behaviors were organized and interconnected. The technical developer (P3) also reported the need to consult the documentation and communicate with the framework's developer to resolve specific technical issues.

System Configuration and Hardware Integration. Responses also included reports of challenges associated with hardware integration and system configuration. In the GRACE project, the issues reported involved virtual machine environments, microphone setup, configuration parameters, and system performance. In the FRED MX project, reported challenges included communication between the ESP32 controller and the host computer,

as well as hardware disconnections associated with robot movement. These observations were reported during the implementation and operation of the robotic platforms.

7.2.4 Discussion

Regarding research question 1 ("RQ1: How did researchers experience the use of the framework in their projects?"), the responses indicate that the researchers were able to use the framework in their projects, despite having different technical backgrounds and project requirements. The reported learning curve ranged from neutral to very easy, while all participants indicated needing a moderate level of assistance while using the framework. These observations suggest that the framework can be learned and used by researchers working on social robotics projects. However, the support required in all cases highlights opportunities to improve the documentation to better guide future users. An interesting observation is that two of the three participants (P2 and P3) reported feeling comfortable guiding other researchers interested in using the framework, whereas the third participant (P1) expressed only partial confidence. Although this result should not be generalized due to the small number of participants, it suggests that researchers who successfully complete a project using the framework can gain sufficient familiarity to support future users.

Regarding research question 2 ("RQ2: How was the framework adapted or extended to meet project requirements?"), the analyzed cases demonstrate various ways the framework was utilized, ranging from adapting existing functionalities to developing entirely new capabilities. The GRACE project required modifications to the robot's expressions and voice characteristics, as well as the integration of a large language model to support conversational interactions. In contrast, the FRED MX research project relied primarily on adapting existing affective and conversational behaviors, while technical development activities included implementing a dedicated control panel to facilitate robot operation. These cases illustrate that the framework was employed in projects with distinct objectives, technical requirements, and levels of customization. The reported modifications also show that researchers were able to leverage existing components rather than developing complete solutions from scratch. Although the degree of customization varied across projects, the framework enabled both the reuse of existing functionalities and the incorporation of new capabilities.

Regarding research question 3 ("RQ3: What strengths and challenges were encountered during framework use?"), the benefits identified were directly associated with re-

duced development effort, the reuse of existing components, and the ability to focus on research objectives rather than low-level implementation tasks. Researchers highlighted the availability of hardware and software infrastructure, pre-implemented behaviors, and support tools as factors that facilitated project development. At the same time, responses revealed recurring challenges related to documentation, the framework's understandability, system configuration, and hardware integration. Difficulties in understanding the software architecture, identifying the responsibilities of different modules, and configuring system components were reported in several cases. Furthermore, practical challenges associated with communication infrastructure and hardware deployment were observed in both projects. Although these challenges did not prevent the projects from being completed, they point to areas for improvement in future versions of the framework. In particular, improved documentation and a better description of the architecture could reduce the effort required when using the framework.

7.2.5 Conclusion

This evaluation analyzed the use of the proposed framework in real-world Human-Robot Interaction (HRI) research projects involving researchers from two institutions. Through an analysis of two research projects utilizing the GRACE and FRED MX robots, the study examined the researchers' experiences with the framework, the modifications made to meet project requirements, and the benefits and challenges associated with its use.

The results indicate that the framework was successfully employed in projects with distinct objectives, application domains, and technical requirements. Researchers reported that the framework reduced development effort by providing reusable hardware and software components, allowing project activities to focus on specific research goals rather than low-level implementation tasks. The reported experiences also demonstrate that the framework enabled varying levels of customization, including the adaptation of existing behaviors, the development of support tools, and the integration of new functionalities.

At the same time, the evaluation identified challenges related to documentation, understanding the framework, configuration procedures, and hardware integration. Although these issues did not prevent the successful completion of the projects, they highlight opportunities to improve the framework and facilitate its use in future research initiatives.

Given the exploratory nature of the study and the limited number of participants, the findings should not be interpreted as statistically generalizable. However, the re-

ported experiences provide evidence that the framework can support the development and implementation of social robot applications across different research contexts.

7.3 Final Remarks

This chapter presented an evaluation of the EvaML 2.0 programming language and the EvaSIM 2.0 simulator, both contributions of this thesis, resulting from the extension of the original versions of these tools. They support the programming, testing, simulation, and execution of interaction scripts for the EVA robot. The chapter investigated the user experience regarding EvaML 2.0 and the usability of the EvaSIM 2.0 simulator in a hackathon-style learning environment, involving participants with little or no prior programming experience. Furthermore, an exploratory evaluation based on case studies was presented, focusing on instantiations of the GRACE and FRED MX robots, developed in Switzerland and Mexico, respectively, carried out by researchers in the field of HRI.

The next chapter concludes this thesis by revisiting the research questions, summarizing the main contributions and findings, and presenting directions for future work.

8 Conclusions

Developing affective social robots involves giving the robot capabilities that allow it to estimate the user’s affective state and transmit its own affective states in an understandable way. This thesis addressed these challenges by proposing a modular and extensible robotic framework featuring a layer dedicated to affective capabilities. This layer provides abstractions for estimating the user’s affective state, modeling and adapting the robot’s affective state, and mapping that state to multimodal behavior specifications.

This chapter presents the conclusions of this thesis. Initially, the research questions and the evidence obtained to answer them are discussed. Next, the main contributions of the study are summarized. Finally, its limitations are discussed and perspectives for future research are presented.

8.1 Answering the Research Questions

The research questions defined at the beginning of this thesis guided the investigation of two main aspects of affective human-robot interaction: how the robot can convey its affective states and how the user’s affective state can be estimated.

How can a social robot convey its affective states in a way that is understandable to users? In this thesis, this question was addressed by combining various non-verbal communication elements available on the FRED robotic platform, including facial expressions, body poses and movements, and RGB LED lighting patterns. Combining these elements enabled the robot to convey different affective states and intensity levels. The evaluation of FRED provided evidence regarding how well participants understood its affective states. Results from the PEA metric showed the highest correspondence with expected responses for happiness (84.81%) and anger (76.4%), followed by surprise (62.68%) and sadness (49.71%). The results also showed that, in some situations, although participants identified the emotion being expressed, they had difficulty correctly distinguishing its level of intensity. These results indicate that the understanding of a robot’s affective states

depends not only on the availability of different communication elements but also on how these elements are combined to represent each state and its intensity levels. Thus, in response to research question 1, the results suggest that effectively conveying an affective state depends not only on the use of multiple modalities, but also on how these modalities are combined to make each state understandable to users.

How can a social robot estimate the user's affective state? In this thesis, this question was addressed through PULSE, a component of the Affective Layer that provides abstractions for processing and integrating affective information from different sources. PULSE defines abstractions for estimating the confidence of incoming information, converting categorical representations into the Valence-Arousal-Dominance (VAD) space, and performing multimodal fusion. The Affective Perception Window (APW) allows for the definition of which perception sources will be used in the estimation and, by specifying their start and end points, the delimitation of the interval during which the information produced by these sources will be considered. To demonstrate this approach, PULSE's extension points were implemented using specific strategies and applied in two affective interaction scenarios involving the robot FRED. In these instantiations, affective information from different sources was converted into a common VAD representation and fused based on confidence levels, yielding an estimate of the user's affective state. The scenarios serve as a functional demonstration of this approach during interaction with the robot, although they did not include an experimental evaluation of accuracy in this process. Thus, in response to research question 2, this thesis proposes an extensible approach for estimating the user's affective state, based on abstractions that allow for the integration of affective information from diverse sources, representations, and strategies for processing and multimodal fusion.

8.2 Contributions

In addition to answering the research questions, this thesis resulted in a set of contributions related to software architecture for affective social robots, affective information processing, and the development of platforms and tools for HRI research. Main contributions include the Social Robotics Framework, the Affective Layer, and the FRED robotic platform. Other contributions resulted from the extension of platforms and tools previously developed in the context of the EVA project.

8.2.1 Social Robotics Framework

One of the main contributions of this thesis is a Social Robotics Framework, a modular and extensible architecture organized into three layers, the Robot Abstraction Layer, the Execution Layer, and the Affective Layer, which decouples platform-specific characteristics from execution mechanisms and affective processing functionalities.

The applicability of the architecture was demonstrated by its implementation and reuse in different contexts. The abstractions defined by RAL were instantiated in EVA 2.0, FRED, GRACE, and FRED MX, platforms with different hardware and software characteristics, and by reusing the Execution Layer’s SENSEI component as a Script Player in the EvaPlay and EVA Virtual applications. These cases provide examples of using different parts of the framework across different platforms and applications. These findings were complemented by a qualitative exploratory assessment involving researchers who developed GRACE and FRED MX. They reported benefits regarding the reuse and adaptation of framework components, alongside challenges related to understanding the architecture, documentation, configuration, and integration. Although the results cannot be generalized, they provide further evidence of the proposed architecture’s utility in independently developed HRI projects.

8.2.2 Affective Layer

Another main contribution of this thesis is the Affective Layer, which organizes, into three complementary components (PULSE, ROSE, and AEB), abstractions related to estimating the user’s affective state, modeling and adapting the robot’s affective state, and mapping that state to multimodal behavior specifications. These components establish a flow between these distinct responsibilities while keeping them separate within the architecture.

The Affective Layer was designed to be extensible, defining extension points that allow for the implementation of various affective processing strategies without altering the layer’s overall organization. In this thesis, these extension points were realized through specific strategies for PULSE, ROSE, and AEB. The resulting instantiation was integrated into FRED and used in two affective interaction scenarios featuring different Robot Affective Profiles. These scenarios functionally demonstrated the integration of the three components and the flow connecting user affective state estimation, the updating of the robot’s affective state, and the mapping of that state to multimodal behavior. Although

they do not constitute an experimental evaluation of the strategies' efficacy, they provide evidence of the presented instantiation's integrated operation and the feasibility of implementing the abstractions defined by the Affective Layer.

8.2.3 FRED Affective Social Robotics Platform

Another main contribution of this thesis is the FRED robot, an open-source, low-cost affective social robotics platform developed to support human–robot interaction research in healthcare and education applications. FRED uses facial expressions, poses, body movements, speech, and the LEDs on its chest to convey its affective state, and its control system implements the abstractions defined by the Robot Abstraction Layer, serving as an instantiation of the proposed framework.

FRED was evaluated by 113 participants regarding its ability to convey affective states and the platform's perceived characteristics. In addition to the results concerning the understanding of affective states, discussed under research question 1, the Godspeed Questionnaire Series assessment yielded a high score for Likeability ($M_d = 5$), a moderately positive rating for Perceived Intelligence ($M_d = 4$), and neutral ratings for Anthropomorphism ($M_d = 3$) and Animacy ($M_d = 3.5$). The Perceived Child Acceptance metric also reached the maximum value ($M_d = 5$), indicating strong agreement that children would enjoy interacting with FRED. Qualitative data complemented these findings, identifying perceptions and opportunities for improvement regarding facial expressions, body movements, appearance, and, most notably, speech. Thus, the evaluation provided evidence regarding both FRED's capacity to convey affective states and the various platform characteristics perceived by the participants.

8.2.4 Secondary Contributions

In addition to the primary contributions, this thesis presents secondary contributions resulting from the evolution of platforms and tools developed within the context of the initial EVA robot project. These contributions include EVA 2.0, EvaML 2.0, and EvaSIM 2.0, which were extended to incorporate new capabilities and align with the architecture proposed in this thesis.

The new version 2.0 of the EVA robot has gained new multimodal interaction capabilities, and its control software has been refactored using the abstractions of the Robot Abstraction Layer, constituting an instantiation of that framework layer.

The EvaML language was extended to version 2.0 to accommodate and support the new interaction capabilities of robotic platforms as well as the new features and requirements of the proposed framework. Its evaluation with novice users via the UEQ-S during an educational hackathon showed positive user experience results. Although variations were observed following the participants' practical experience, the differences between the evaluation stages were not statistically significant. These results provide favorable evidence regarding the experience of language use in the investigated context, without indicating a statistically demonstrated improvement throughout the experiment.

Refactoring EvaSIM to version 2.0 expanded the simulator's functionality, enabling it to also serve as a script player for the physical robot. Its evaluation returned a mean SUS score of 66.25, corresponding to the "Marginally Acceptable" range and an "OK" rating, while also revealing several opportunities for improvement, primarily regarding ease of learning and the perceived need for technical support.

8.3 Limitations

Despite the contributions and results presented, this work has limitations that must be considered when interpreting its conclusions. Regarding the Affective Layer, the abstractions and extension points defined in PULSE, ROSE, and AEB were implemented and integrated into the FRED robot. The functioning of this integration was demonstrated through two affective interaction scenarios. Although this demonstration provides functional evidence of the proposed instantiation, it is limited to the scenarios and strategies implemented in this work, preventing an assessment of the architecture's behavior when instantiated with other strategies, platforms, or interaction contexts.

The evaluation of FRED's capabilities for conveying affective states also has limitations related to the experimental design. The use of videos allowed for standardization of stimuli, but prevented direct physical interaction with the robot. Furthermore, in some blocks, the previous affective expression persisted at the start of the next stimulus, potentially influencing the perception of surprise and sadness. Because the phrases used in the dialogue between the robots did not cover all of FRED's affective states, the states of "Mild Anger" and "Intense Sadness" were not presented and, consequently, could not be evaluated.

The evaluation of EvaML 2.0 and EvaSIM 2.0 relied on a relatively small sample within the specific context of a hackathon. This limits the statistical power of the analyses and

the validity of the results. Thus, the findings should be interpreted within the investigated context and should not be directly generalized to other users or usage contexts.

Finally, the assessment of the framework's applicability was qualitative and exploratory in nature, involving three researchers from two independent projects, GRACE and FRED MX, carried out at other institutions. Although these cases provide external evidence on the use and adaptation of framework components, the small number of participants and projects limits generalization to other researchers, platforms, and HRI contexts.

8.4 Future Work

The results and limitations presented in this thesis point to various possibilities for future work. Regarding the Affective Layer, new instantiations could explore and compare different strategies at the extension points of PULSE, ROSE, and AEB, as well as investigate their use and integration within other contexts of the AHRI. Specifically for AEB, we intend to instantiate the multimodal mapping using the Behavior Markup Language (BML) for the specification and coordination of multimodal behaviors.

Regarding FRED, we propose refining and re-evaluating the affective states that proved difficult to recognize or distinguish in terms of intensity, as well as conducting evaluations involving direct physical interaction. From an aesthetic standpoint, the aim is to improve the robot's design and upgrade its more limited electronic components, such as the 8x8 LED arrays used to represent its eyes and mouth. Furthermore, we intend to utilize and investigate the FRED platform in interactions with children and within educational settings.

Regarding EvaML 2.0 and EvaSIM 2.0, evaluations involving larger samples and different contexts could expand upon the evidence gathered during the hackathon. Future versions of EvaSIM 2.0 might also address aspects identified by the SUS, specifically ease of learning and the perceived need for technical support.

Finally, it is intended is to address the requirements identified by researchers from the GRACE and FRED MX projects, seeking to resolve challenges related to understanding the architecture, documentation, configuration, and integration. Furthermore, the aim is to evaluate the structure involving additional independent researchers, projects, and robotic platforms, thereby generating more comprehensive evidence regarding the architecture's applicability and extensibility across various human-robot interaction (HRI) contexts.

8.5 Publications

During the development of this thesis, the following papers were published in journals:

- Rocha, Marcelo; Favela, J.; Muchaluat-Saade, D. C. (2025). **Design and evaluation of an XML-based language for programming socially assistive robots.** INTERNATIONAL JOURNAL OF SOCIAL ROBOTICS, 17(7), 1305–1333.
- Rocha, Marcelo; Cruz-Sandoval, Dagoberto; Favela, Jesus; Muchaluat-Saade, Débora. **Design and usability evaluation of the EvaSIM simulator for a socially assistive robot.** MULTIMEDIA TOOLS AND APPLICATIONS. 2024
- Favela, Jesus.; Dagoberto Cruz-Sandoval; Rocha, Marcelo; Débora Muchaluat-Saade. **Social Robots for Healthcare and Education in Latin America.** COMMUNICATIONS OF THE ACM. v.67, p.70 - 71, 2024.
- Souza, T. G.; Seixas, E. F. R.; Rocha, Marcelo; A. B.; Eller, E. N. P.; Cabral, S. F.; Lacerda, I. R. G.; Nascimento, A. de P.; Muchaluat-Saade, D. C.; Salgado, L. C. de C. **Designing Situated Computing Experiences for Girls: Intersectionality, CTScale, and HCI Artifacts in Hackathon 360.** JOURNAL ON INTERACTIVE SYSTEMS, Porto Alegre, RS, v. 17, n. 1, p. 903–925, 2026. DOI: 10.5753/jis.2026.7538.

The following paper is under review:

- **Towards an Affective Robot: Evaluating FRED's Affective Expressions,** submitted to the INTERNATIONAL JOURNAL OF SOCIAL ROBOTICS.

The following papers were published in conferences:

- Rocha, Marcelo; Favela, Jesus; Muchaluat-Saade, Débora C. **Uma Proposta de Framework para Sistemas de Software de Controle para Plataformas de Robótica Social.** In: Proceedings of the Brazilian Symposium on Multimedia and the Web, 2024, Brasil. Proceedings of the 30th Brazilian Symposium on Multimedia and the Web (WebMedia 2024). 2024, p.327

- Rocha, Marcelo; Muchaluat-Saade, Débora C. **Friendly Robot for Education and Healthcare: FRED**. In: LIQUE 2023 - Life Improvement in Quality by Ubiquitous Experiences Workshop, 2023, Nates France. ACM International Conference on Interactive Media Experiences Workshop, 2023.
- Rocha, Marcelo; Conci, Aura; Muchaluat-Saade, Débora **Voice Emotion Recognition Based on Color Histogram Features**. In: 2023 IEEE 36th International Symposium on ComputerBased Medical Systems (CBMS), 2023, L'Aquila. 2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS). 2023, p.341
- Rocha, Marcelo; Muchaluat-Saade, Débora; Clua, Esteban **A 3D Simulator for a Social and Ludic Robotics Platform**. In: Simpósio Brasileiro de Jogos e Entretenimento Digital - SBGames23, 2023, Rio Grande. Anais do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2023)
- Rocha, Marcelo; Muchaluat-Saade, Débora C. **EvaML e EvaSIM: Proposta de Linguagem Baseada em XML e Simulador para o Robô EVA**. In: O Concurso de Teses e Dissertações (CTD) do WebMedia, 2023, Ribeirão Preto, SP. [Anais Estendidos do XXIX Simpósio Brasileiro de Sistemas Multimídia e Web, 2023.
- Rocha, Marcelo; Débora Muchaluat-Saade. **Embodied Voice Assistant Markup Language**. In: Proceedings of the Brazilian Symposium on Multimedia and the Web, 2023, Brasil. Proceedings of the 29th Brazilian Symposium on Multimedia and the Web (WebMedia 2023), 2023, Ribeirão Preto, SP.
- Rocha, Marcelo, Elaine F. Rangel Seixas, Flavio Seixas, and Débora C. Muchaluat-Saade. **Desafios da Robótica Socialmente Assistiva para Idosos e Pessoas com TEA**. Grandes Desafios da Computação Aplicada à Saúde. Sociedade Brasileira de Computação, 2026, pp. 110–121.
- Rocha, Marcelo; Muchaluat-Saade, Débora C. **EvaML e EvaSIM: Proposta de Linguagem Baseada em XML e Simulador para o Robô EVA**. In: Concurso de Teses e Dissertações, 2023, Brasil. Anais do XXXVI Concurso de Teses e Dissertações (CTD 2023). p. 98.
- Rangel, Cristina I.; Morales-Téllez, Arturo; Rocha, Marcelo; Muchaluat-Saade, Débora C.; Favela, Jesús. **A Social Affective Robot for Emotion Regulation to Mitigate Procrastination in Adolescents**. Lecture Notes in Networks and Systems. 1ed.: Springer Nature Switzerland, 2026, v. , p. 33-44

- Silva, Pedro Lucas Pereira da; Cecim, João Vitor Carvalho; Rocha, Marcelo; Muchaluat-Saade, Débora C. **Serious Games to Support Cognitive Exercise in Older Adults Using the Social Robot EVA**. In: Proceedings of the ACM International Conference on Interactive Media Experiences Workshops (IMXw). SBC, 2025. p. 87-90.
- Vieira, R.; Rocha, M.; Albuquerque, C.; Muchaluat-Saade, D. C.; César, P. **Remix-drum: A smart musical instrument for music and visual art remix**. In 2023 IEEE 9th World Forum on Internet of Things (WF-IoT) (pp. 1-7). IEEE.
- Moreira, Pedro Castelo Branco, et al. **Design and Evaluation of a Mobile Assistive Robot Simulator for Therapy Continuity in ASD Care**. 2026 IEEE Conference on Serious Games and Applications for Health (SeGAH). IEEE, 2026.
- Silva, Pedro Lucas Pereira da; Cecim, João Vitor Carvalho; Rocha, Marcelo; Muchaluat-Saade, Débora C. **Jogos Sérios para Auxiliar o Exercício Cognitivo de Idosos Utilizando o Robô Social EVA**. In: Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web, 2024, Brasil. Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2024). p. 95.

REFERENCES

- AHMAD, Muneeb I; KHORDI-MOODI, Mark; LOHAN, Katrin S. Social robot for stem education. In: COMPANION of the 2020 ACM/IEEE International Conference on Human-Robot Interaction. [S.l.: s.n.], 2020. P. 90–92.
- ALBONICO, Michel; ĐORĐEVIĆ, Milica; HAMER, Engel; MALAVOLTA, Ivano. Software engineering research on the robot operating system: a systematic mapping study. **Journal of systems and software**, Elsevier, v. 197, p. 111574, 2023.
- ALI, Sara; MEHMOOD, Faisal; IQBAL, Khawaja Fahad; AYAZ, Yasar; SAJID, Muhammad; SIAL, Muhammad Baber; MALIK, Muhammad Faiq; JAVED, Kashif. Human robot interaction: identifying resembling emotions using dynamic body gestures of robot. In: IEEE. 2023 3rd International Conference on Artificial Intelligence (ICAI). [S.l.: s.n.], 2023. P. 39–44.
- ALSWAIDAN, Nourah; MENAI, Mohamed El Bachir. A survey of state-of-the-art approaches for emotion recognition in text. **Knowledge and information systems**, Springer, v. 62, n. 8, p. 2937–2987, 2020.
- ASHOK, Ashita; PAPLU, Sarwar; BERNIS, Karsten. Social robot dressing style: an evaluation of interlocutor preference for university setting. In: IEEE. 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). [S.l.: s.n.], 2023. P. 2499–2504.
- BAILLIE, J-C. Urbi: towards a universal robotic body interface. In: 4TH IEEE/RAS International Conference on Humanoid Robots, 2004. [S.l.: s.n.], 2004. v. 1, 33–51 vol. 1. DOI: [10.1109/ICHR.2004.1442112](https://doi.org/10.1109/ICHR.2004.1442112).
- BAILLIE, J. C. Urbi: towards a universal robotic low-level programming language. In: IEEE. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l.: s.n.], 2005. P. 820–825.
- BAL, Fatma; TEKEREK, Mehmet; PALACZ, Magdalena; GÖK, Mehmet; ŞİMŞİR, Ramazan. Human robot interaction with social humanoid robots. **El-cezeri**, Tayfun UYGUNOĞLU, v. 11, n. 1, p. 94–102, 2024.

BALAJI, Atul. **Emotion detection**. [S.l.: s.n.], 2020.

<https://github.com/atulapra/Emotion-detection>. GitHub repository. Accessed: September 17, 2026.

BANGOR, Aaron; KORTUM, Philip T; MILLER, James T. An empirical evaluation of the system usability scale. **Intl. journal of human–computer interaction**, Taylor & Francis, v. 24, n. 6, p. 574–594, 2008. DOI:

<https://psycnet.apa.org/doi/10.1080/10447310802205776>.

BARTNECK, Christoph; KULIĆ, Dana; CROFT, Elizabeth; ZOGHBI, Susana.

Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. **International journal of social robotics**, Springer, v. 1, p. 71–81, 2009.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software architecture in practice**. 4th. [S.l.]: Addison-Wesley Professional, 2021. (SEI Series in Software Engineering). ISBN 978-0136886099.

BECK, Aryel; CAÑAMERO, Lola; HIOLE, Antoine; DAMIANO, Luisa; COSI, Piero; TESSER, Fabio; SOMMAVILLA, Giacomo. Interpretation of emotional body language displayed by a humanoid robot: a case study with children. **International journal of social robotics**, Springer, v. 5, p. 325–334, 2013.

BELPAEME, Tony; KENNEDY, James; RAMACHANDRAN, Aditi;

SCASSELLATI, Brian; TANAKA, Fumihide. Social robots for education: a review.

Science robotics, American Association for the Advancement of Science, v. 3, n. 21, eaat5954, 2018.

BERTOL, Douglas Wildgrube; FINK, Geoff; MARCHITTO, Marco; NISTICÓ, Ylenia; PESTARINO, Michele; SEMINI, Claudio. Dls2: a distributed microcomponent-based robotic software architecture. **Results in engineering**, Elsevier, p. 112048, 2026.

BIANCHI, Renata Germano; ALMEIDA NERIS, Vânia Paula de. Which color should i pick? a study on colors and emotions in human-computer interaction. In:

PROCEEDINGS of the 14th Brazilian Symposium on Human Factors in Computing Systems. [S.l.: s.n.], 2015. P. 1–10.

BISCHOFF, Rainer; GRAEFE, Volker. Dependable multimodal communication and interaction with robotic assistants. In: IEEE. PROCEEDINGS. 11th IEEE International Workshop on Robot and Human Interactive Communication. [S.l.: s.n.], 2002.

P. 300–305.

BRADSKI, Gary. The opencv library. **Dr. dobb's journal of software tools**, 2000.

- BRAVO PERUCHO, Aurea; ALIMARDANI, Maryam. Social robots in secondary education: can robots assist young adult learners with math learning? In: COMPANION of the 2023 ACM/IEEE international conference on human-robot interaction. [S.l.: s.n.], 2023. P. 355–359.
- BRETSCHNEIDER, Maximilian; MANDL, Sarah; STROBEL, Anja; ASBROCK, Frank; MEYER, Bertolt. Social perception of embodied digital technologies—a closer look at bionics and social robotics. **Gruppe. interaktion. organisation. zeitschrift für angewandte organisationspsychologie (gio)**, Springer, v. 53, n. 3, p. 343–358, 2022.
- BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans; SOMMERLAD, Peter; STAL, Michael. **Pattern-oriented software architecture, volume 1: a system of patterns**. [S.l.]: John Wiley & Sons, 1996.
- CALDIERA, Victor R Basili1 Gianluigi; ROMBACH, H Dieter. The goal question metric approach. **Encyclopedia of software engineering**, p. 528–532, 1994.
- CANO, Sandra; HIDALGO, Rocio; PELÁEZ, Carlos Alberto; SOLANO, Andrés; GUZMÁN, Francisca; REYES, Ignacio; VILLALBA-CONDORI, Klinge Orlando. A child–robot interaction framework for designing inclusive educational robotics in resourced classrooms:(robo-icare). **Applied sciences**, MDPI, v. 16, n. 4, p. 1862, 2026.
- CAO, Hoang-Long; VAN DE PERRE, Greet; SIMUT, Ramona; POP, Cristina; PECA, Andreea; LEFEBER, Dirk; VANDERBORGHT, Bram. Enhancing my keepon robot: a simple and low-cost solution for robot platform in human-robot interaction studies. In: IEEE. THE 23rd IEEE International Symposium on Robot and Human Interactive Communication. [S.l.: s.n.], 2014. P. 555–560.
- CÁRDENAS, Pedro; GARCÍA, José; BEGAZO, Rolinson; AGUILERA, Ana; DONGO, Irvin; CARDINALE, Yudith. Evaluation of robot emotion expressions for human–robot interaction. **International journal of social robotics**, Springer, p. 1–23, 2024.
- CHITA-TEGMARK, Meia; LOHANI, Monika; SCHEUTZ, Matthias. Gender effects in perceptions of robots and humans with varying emotional intelligence. In: IEEE. 2019 14th acm/IEEE international conference on human-robot interaction (hri). [S.l.: s.n.], 2019. P. 230–238.
- CONROY, Ronán M. What hypotheses do “nonparametric” two-group tests actually test? **The stata journal**, SAGE Publications Sage CA: Los Angeles, CA, v. 12, n. 2, p. 182–190, 2012.

CRUZ SANDOVAL, D. **Robot conversacional como apoyo a intervenciones no farmacológicas para adultos mayores con demencia conversational robot to support non-pharmacological interventions for people with dementia**. 2020.

Tesis de Doctorado en Ciencias – Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. 125pp. Available from:

<<http://cicese.repositorioinstitucional.mx/jspui/handle/1007/3283>>.

CRUZ-SANDOVAL, Dagoberto; FAVELA, Jesus. A Conversational Robot to Conduct Therapeutic Interventions for Dementia. **Ieee pervasive computing**, v. 18, n. 2, p. 10–19, 2019. ISSN 15582590. DOI: [10.1109/MPRV.2019.2907020](https://doi.org/10.1109/MPRV.2019.2907020).

DARDA, Kohinoor M; MAIWALD, Aaron; RAGHURAM, Tanvi; CROSS, Emily S. Dancing robots: aesthetic engagement is shaped by stimulus and knowledge cues to human animacy. **Frontiers in human neuroscience**, Frontiers Media SA, v. 18, p. 1413066, 2024.

DE BRUYNE, Luna; DE CLERCQ, Orphée; HOSTE, Veronique. An emotional mess! deciding on a framework for building a dutch emotion-annotated corpus. In: PROCEEDINGS of the Twelfth Language Resources and Evaluation Conference. [S.l.: s.n.], 2020. P. 1643–1651.

DÉSIRON, Juliette C; SCHNEIDER, Sascha. Exploring the interplay of information relevance and colorfulness in multimedia learning. **Frontiers in psychology**, Frontiers Media SA, v. 15, p. 1393113, 2024.

DZHOROEV, Temirlan; PARK, Haeun; LEE, Jiyeon; KIM, Byounghern; LEE, Hui Sung. Human perception on social robot's face and color expression using computational emotion model. In: IEEE. 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). [S.l.: s.n.], 2023. P. 2484–2491.

EKMAN, Paul. An argument for basic emotions. **Cognition & emotion**, Taylor & Francis, v. 6, n. 3-4, p. 169–200, 1992.

EKSTRÖM, Sara; PARETO, Lena. The dual role of humanoid robots in education: as didactic tools and social actors. **Education and information technologies**, Springer, v. 27, n. 9, p. 12609–12644, 2022.

EMBGEN, Stephanie; LUBER, Matthias; BECKER-ASANO, Christian; RAGNI, Marco; EVERS, Vanessa; ARRAS, Kai O. Robot-specific social cues in emotional body language. In: IEEE. 2012 IEEE RO-MAN: The 21st IEEE International

Symposium on Robot and Human Interactive Communication. [S.l.: s.n.], 2012. P. 1019–1025.

ERDEN, Mustafa Suphi. Emotional postures for the humanoid-robot nao.

International journal of social robotics, Springer, v. 5, p. 441–456, 2013.

FAISAL, Mohammed; ALHARBI, Abdullah; ALHAMADI, Amnah;

ALMUTAIRI, Sarah; ALENEZI, Shaikhah; ALSULAILI, Anfal; KHAN, Murad;

KHAN, Faheem. Robot-based solution for helping alzheimer patients. **Slas technology**, Elsevier, v. 29, n. 3, p. 100140, 2024.

FERNÁNDEZ-RODICIO, Enrique; MAROTO-GÓMEZ, Marcos;

CASTRO-GONZÁLEZ, Álvaro; MALFAZ, María; SALICHS, Miguel Á. Emotion and mood blending in embodied artificial agents: expressing affective states in the mini social robot. **International journal of social robotics**, Springer, v. 14, n. 8, p. 1841–1864, 2022.

FERRARI, Francesco; EYSSEL, Friederike. Toward a hybrid society: the transformation of robots, from objects to social agents. In: SPRINGER. INTERNATIONAL Conference on Social Robotics. [S.l.: s.n.], 2016. P. 909–918.

FOWLER, Martin. **Inversion of control containers and the dependency injection pattern**. [S.l.: s.n.], 2004. <https://martinfowler.com>.

FRAGAKIS, Nikos; TRICHOPOULOS, Georgios; CARIDAKIS, George. Empowering education with intelligent systems: exploring llms and the nao robot for information retrieval, 2025.

FREEMAN, Eric; ROBSON, Elisabeth. **Head first design patterns: building extensible and maintainable object-oriented software**. 2nd. [S.l.]: O'Reilly Media, 2020. ISBN 978-1492078005.

FUGLERUD, Kristin S; SOLHEIM, Ivar. The use of social robots for supporting language training of children. In: TRANSFORMING Our World Through Design, Diversity And Education. [S.l.]: IOS Press, 2018. P. 401–408.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Design patterns: elements of reusable object-oriented software**. Reading, MA: Addison-Wesley, 1994.

GAO, Wa; SHEN, Shiyi; JI, Yang; TIAN, Yuan. Human perception of the emotional expressions of humanoid robot body movements: evidence from survey and eye-tracking measurements. **Biomimetics**, MDPI, v. 9, n. 11, p. 684, 2024.

- GARCIA, Sergio; GOMEZ-DONOSO, Francisco; CAZORLA, Miguel. Enhancing human–robot interaction: development of multimodal robotic assistant for user emotion recognition. **Applied sciences**, MDPI, v. 14, n. 24, p. 11914, 2024.
- GEORGE, Darren; MALLERY, Paul. **Ibm spss statistics 26 step by step: a simple guide and reference**. [S.l.]: Routledge, 2019.
- GOODFELLOW, Ian J.; ERHAN, Dumitru; CARRIER, Pierre Luc; COURVILLE, Aaron; MIRZA, Mehdi; HAMNER, Ben; CUKIERSKI, Will; TANG, Yichuan; THALER, David; LEE, Dong-Hyun; ZHOU, Yingbo; RAMAIAH, Chetan; FENG, Fangxiang; LI, Ruifan; WANG, Xiaojie; ATHANASAKIS, Dimitris; SHAW-TAYLOR, John; MILAKOV, Maxim; PARK, John; IONESCU, Radu; POPESCU, Marius; GROZEA, Cristian; BERGSTRA, James; XIE, Jingjing; ROMASZKO, Lukasz; XU, Bing; CHUANG, Zhang; BENGIO, Yoshua. Challenges in representation learning: a report on three machine learning contests. In: **NEURAL Information Processing**. [S.l.]: Springer, 2013. v. 8228. (Lecture Notes in Computer Science), p. 117–124. DOI: [10.1007/978-3-642-42051-1_16](https://doi.org/10.1007/978-3-642-42051-1_16).
- HÄRING, Markus; BEE, Nikolaus; ANDRÉ, Elisabeth. Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots. In: **IEEE. 2011 RO-MAN**. [S.l.: s.n.], 2011. P. 204–209.
- HART, Anna. Mann-whitney test is not just a test of medians: differences in spread can be important. **Bmj**, British Medical Journal Publishing Group, v. 323, n. 7309, p. 391–393, 2001.
- HARTMANN, Jochen. **Emotion english distilroberta-base**. [S.l.: s.n.], 2022. <https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/>.
- HENSCHER, Anna; LABAN, Guy; CROSS, Emily S. What makes a robot social? a review of social robots from fiction to a home or hospital near you. **PsyArXiv**, 2020.
- HOLROYD, Aaron; RICH, Charles. Using the behavior markup language for human-robot interaction. In: **PROCEEDINGS of the seventh annual ACM/IEEE international conference on Human-Robot Interaction**. [S.l.: s.n.], 2012. P. 147–148.
- HÜBERT, Heiko; YUN, Hae Seon. Sobotify: a framework for turning robots into social robots. In: **PROCEEDINGS of the 2024 ACM/IEEE International Conference on Human-Robot Interaction**. [S.l.: s.n.], 2024. P. 910–914.

- JEON, Myounghoon; ZHANG, Ruimin; LEHMAN, William; FAKHRHOSSEINI, Seyedeh; BARNES, Jaclyn; PARK, Chung Hyuk. Development and evaluation of emotional robots for children with autism spectrum disorders. In: SPRINGER. HCI International 2015-Posters' Extended Abstracts: International Conference, HCI International 2015, Los Angeles, CA, USA, August 2–7, 2015. Proceedings, Part I. [S.l.: s.n.], 2015. P. 372–376.
- JIRAK, Doreen; AOKI, Motonobu; YANAGI, Takura; TAKAMATSU, Atsushi; BOUET, Stephane; YAMAMURA, Tomohiro; SANDINI, Giulio; REA, Francesco. Is it me or the robot? a critical evaluation of human affective state recognition in a cognitive task. **Frontiers in neurorobotics**, Frontiers Media SA, v. 16, p. 882483, 2022.
- JOBaida, Nohori; SHIDUJAMAN, Mohammad; MAHBUB, Ahsan; SAMANI, Hooman; PREMACHANDRA, Chinthaka; UDDIN, Mohammad Faisal; ALAM, Saadia Binte. Design and development of a low-cost voice interactive children educational robot 'tiny' with natural language processing. In: IEEE. 2024 International Conference on Image Processing and Robotics (ICIPRoB). [S.l.: s.n.], 2024. P. 1–6.
- KANG, Hangyeol; BEN MOUSSA, Maher; THALMANN, Nadia Magnenat. Nadine: a large language model-driven intelligent social robot with affective capabilities and human-like memory. **Computer animation and virtual worlds**, Wiley Online Library, v. 35, n. 4, e2290, 2024.
- KATO, Misa; OHARA, Kenichi. Robot software framework with different middleware platforms. In: IEEE. 2024 IEEE/SICE International Symposium on System Integration (SII). [S.l.: s.n.], 2024. P. 1433–1438.
- KAYUKAWA, Yuki; TAKAHASHI, Yasutake; TSUJIMOTO, Takuya; TERADA, Kazunori; INOUE, Hiroyuki. Influence of emotional expression of real humanoid robot to human decision-making. In: IEEE. 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE). [S.l.: s.n.], 2017. P. 1–6.
- KAZMAN, Rick; KLEIN, Mark; BARBACCI, Mario; LONGSTAFF, Tom; LIPSON, Howard; CARRIERE, Jeromy. The architecture tradeoff analysis method. In: IEEE. PROCEEDINGS. fourth ieee international conference on engineering of complex computer systems (cat. no. 98ex193). [S.l.: s.n.], 1998. P. 68–78.
- KOPP, Stefan; KRENN, Brigitte; MARSELLA, Stacy; MARSHALL, Andrew N; PELACHAUD, Catherine; PIRKER, Hannes; THÓRISSON, Kristinn R; VILHJÁLMSOHN, Hannes. Towards a common framework for multimodal generation:

the behavior markup language. In: SPRINGER. INTERNATIONAL workshop on intelligent virtual agents. [S.l.: s.n.], 2006. P. 205–217.

KOYAMA, Naoki; TANAKA, Kazuaki; OGAWA, Kohei; ISHIGURO, Hiroshi. Emotional or social? how to enhance human-robot social bonding. In: PROCEEDINGS of the 5th International Conference on Human Agent Interaction. [S.l.: s.n.], 2017. P. 203–211.

KOZIOLEK, Heiko; GRÜNER, Sten; RÜCKERT, Julius. A comparison of mqtt brokers for distributed iot edge computing. In: _____. **Software architecture**. Cham: Springer International Publishing, 2020. P. 352–368. ISBN 978-3-030-58923-3.

LAEEQUE, Bisma; AKMAL, Asma. Empirical evidence of color on motivation toward learning. **International journal of learning and development**, Macrothink Institute, v. 7, n. 2, p. 1–17, 2017.

LEVINSON, Leigh; GVIRSMAN, Omer; GORODESKY, Iris Melamed; PEREZ, Almogit; GONEN, Einat; GORDON, Goren. Learning in summer camp with social robots: a morphological study. **International journal of social robotics**, Springer, v. 13, p. 999–1012, 2021.

LI, Haizhou; JOHN-JOHN, Cabibihan; TAN, Yeow Kee. Towards an effective design of social robots. **International journal of social robotics**, Springer Nature BV, v. 3, n. 4, p. 333–335, 2011.

LI, Liming; ZHAO, Zeang. Designing behaviors of robots based on the artificial emotion expression method in human–robot interactions. **Machines**, MDPI, v. 11, n. 5, p. 533, 2023.

LI, Yuanchao; ISHI, Carlos Toshinori; WARD, Nigel; INOUE, Koji; NAKAMURA, Shizuka; TAKANASHI, Katsuya; KAWAHARA, Tatsuya. Emotion recognition by combining prosody and sentiment analysis for expressing reactive emotion by humanoid robot. In: IEEE. 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). [S.l.: s.n.], 2017. P. 1356–1359.

LÓPEZ, Joaquin; PÉREZ, Diego; ZALAMA, Eduardo. A framework for building mobile single and multi-robot applications. **Robotics and autonomous systems**, Elsevier, v. 59, n. 3-4, p. 151–162, 2011.

- LOPEZ-CAUDANA, Edgar; PONCE, Pedro; MAZON, Nancy; BALTAZAR, German. Improving the attention span of elementary school children for physical education through an nao robotics platform in developed countries. **International journal on interactive design and manufacturing (ijidem)**, Springer, v. 16, n. 2, p. 657–675, 2022.
- MARTIN, Jon; ANSUATEGI, Ander; MAURTUA, Iñaki; GUTIERREZ, Aitor; OBREGÓN, David; CASQUERO, Oskar; MARCOS, Marga. A generic ros-based control architecture for pest inspection and treatment in greenhouses using a mobile manipulator. **Ieee access**, IEEE, v. 9, p. 94981–94995, 2021.
- MENGACCI, Riccardo; ZAMBELLA, Grazia; GRIOLI, Giorgio; CAPORALE, Danilo; CATALANO, Manuel G; BICCHI, Antonio. An open-source ros-gazebo toolbox for simulating robots with compliant actuators. **Frontiers in robotics and ai**, Frontiers Media SA, v. 8, p. 713083, 2021.
- METTA, Giorgio; FITZPATRICK, Paul; NATALE, Lorenzo. Yarp: yet another robot platform. **International journal of advanced robotic systems**, SAGE Publications Sage UK: London, England, v. 3, n. 1, p. 8, 2006.
- MISHRA, Reeta; TRIPATHI, Padmesh; KUMAR, Nitendra. Application of emotional intelligence in improvement of human-robot collaboration. In: HUMAN-MACHINE Collaboration and Emotional Intelligence in Industry 5.0. [S.l.]: IGI Global, 2024. P. 251–267.
- MITJANS, Adrian Acosta. **Affective computation in human-robot interaction**. 2020. Master thesis – Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. Available from: <<http://cicese.repositorioinstitucional.mx/jspui/handle/1007/3283>>.
- MOKHTAR, Tarek H. Designing social robots at scales beyond the humanoid. In: SOCIAL robots: Technological, societal and ethical aspects of human-robot interaction. [S.l.]: Springer, 2019. P. 13–35.
- MOREIRA, Pedro Castelo Branco; SEIXAS, Elaine F Rangel; CECIM, João Vitor Carvalho; SEIXAS, Flavio; MUCHALUAT-SAADE, Débora C; ROCHA, Marcelo Marques da. Design and evaluation of a mobile assistive robot simulator for therapy continuity in asd care. In: IEEE. 2026 IEEE Conference on Serious Games and Applications for Health (SeGAH). [S.l.: s.n.], 2026. P. 1–8.

- MUÑOZ, Luis Miguel. Ergonomic design in interaction with robots. **Journal of ergonomics**, Longdom Publishing, v. 10, n. 1, e186, 2020. ISSN 2165-7556. DOI: [10.35248/2165-7556.20.10.e186](https://doi.org/10.35248/2165-7556.20.10.e186). Available from: <https://www.longdom.org/open-access/ergonomic-design-in-interaction-with-robots-52972.html>.
- MUSICCO, Anna Teresa; TRAETTA, Luigi. Nao and emotional learning: a systematic review of applications in primary schools. **Italian journal of educational technology**, 2026.
- NAGY, Enikő; KARL, Éva; MOLNÁR, György. Exploring the role of human-robot interactions, within the context of the effectiveness of a nao robot, 2024.
- NAMLISESLI, Deniz; BAŞ, Hale Nur; BOSTANCI, Hilal; COŞKUN, Buket; BARKANA, Duygun Erol; TARAKÇI, Devrim. The effect of use of social robot nao on children's motivation and emotional states in special education. In: IEEE. 2024 21st International Conference on Ubiquitous Robots (UR). [S.l.: s.n.], 2024. P. 7–12.
- NEERINCX, Anouk; VELDHUIS, Daan; MASTHOFF, Judith MF; GRAAF, Maartje MA de. Co-designing a social robot for child health care. **International journal of child-computer interaction**, Elsevier, v. 38, p. 100615, 2023.
- OLLAMA INC. **Ollama**. [S.l.], 2026. Versão 0.30.10. Available from: <https://ollama.com>. Visited on: 5 Sept. 2026.
- PANAYIOTOU, Konstantinos; TSARDOULIAS, Emmanouil; ZOLOTAS, Christoforos; SYMEONIDIS, Andreas L; PETROU, Loukas. A framework for rapid robotic application development for citizen developers. **Software**, MDPI, v. 1, n. 1, p. 53–79, 2022.
- PINO, Olimpia; PALESTRA, Giuseppe; TREVINO, Rosalinda; DE CAROLIS, Berardina. The humanoid robot nao as trainer in a memory program for elderly people with mild cognitive impairment. **International journal of social robotics**, Springer, v. 12, p. 21–33, 2020.
- QUIGLEY, Morgan; CONLEY, Ken; GERKEY, Brian; FAUST, Josh; FOOTE, Tully; LEIBS, Jeremy; WHEELER, Rob; NG, Andrew Y, et al. Ros: an open-source robot operating system. In: KOBE, JAPAN, 3.2. ICRA workshop on open source software. [S.l.: s.n.], 2009. v. 3, p. 5.
- RAGGIOLI, Luca; ESPOSITO, Raffaella; ROSSI, Alessandra; ROSSI, Silvia. Exploring the role of robot's movements for a transparent affective communication. **Ieee robotics and automation letters**, IEEE, v. 10, n. 5, p. 4364–4371, 2025.

- RAHMAN, SM Mizanoor. Dynamic affect-based motion planning of a humanoid robot for human-robot collaborative assembly in manufacturing. **Electronics**, MDPI, v. 13, n. 6, p. 1044, 2024.
- RAKHYMBAYEVA, Nazerke; AMIROVA, Aida; SANDYGULOVA, Anara. A long-term engagement with a social robot for autism therapy. **Frontiers in robotics and ai**, Frontiers Media SA, v. 8, p. 669972, 2021.
- RANGEL, Cristina I; MORALES-TÉLLEZ, Arturo; ROCHA, Marcelo Marques da; MUCHALUAT-SAADE, Débora C; FAVELA, Jesús. A social affective robot for emotion regulation to mitigate procrastination in adolescents. In: SPRINGER. INTERNATIONAL Conference on Ubiquitous Computing and Ambient Intelligence. [S.l.: s.n.], 2025. P. 33–44.
- READ, Robin; BELPAEME, Tony. People interpret robotic non-linguistic utterances categorically. **International journal of social robotics**, Springer, v. 8, p. 31–50, 2016.
- RICHARDS, Mark; FORD, Neal. **Fundamentals of software architecture: an engineering approach**. [S.l.]: O'Reilly Media, 2020. ISBN 978-1492043454.
- RITSCHHEL, Hannes. Real-time generation and adaptation of social companion robot behaviors, 2022.
- RITSCHHEL, Hannes; KIDERLE, Thomas; ANDRÉ, Elisabeth. Implementing parallel and independent movements for a social robot's affective expressions. In: IEEE. 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). [S.l.: s.n.], 2021. P. 1–4.
- ROCHA, Marcelo. **Evaml e evasim: proposta de linguagem baseada em xml e simulador para o robô eva**. 2022. Dissertação (Mestrado em Nome do Programa Pós Graduação em Computação da UFF) – Universidade Federal Fluminense, Niterói, RJ.
- ROCHA, Marcelo; CRUZ-SANDOVAL, Dagoberto; FAVELA, Jesus; MUCHALUAT-SAADE, Débora C. Design and usability evaluation of the evasim simulator for a socially assistive robot. **Multimedia tools and applications**, Springer, p. 1–26, 2024.
- ROCHA, Marcelo; VALENTIM, Pedro; BARRETO, Fábio; MITJANS, Adrian; CRUZ-SANDOVAL, Dagoberto; FAVELA, Jesus; MUCHALUAT-SAADE, Débora. Towards enhancing the multimodal interaction of a social robot to assist children with autism in emotion regulation. In: SPRINGER. INTERNATIONAL Conference on Pervasive Computing Technologies for Healthcare. [S.l.: s.n.], 2022. P. 398–415.

- ROCHA, Marcelo Marques da; FAVELA, Jesus; MUCHALUAT-SAADE, Débora C. Design and evaluation of an xml-based language for programming socially assistive robots. **International journal of social robotics**, Springer, v. 17, n. 7, p. 1305–1333, 2025.
- ROCHA, Marcelo Marques Da; MUCHALUAT-SAADE, Débora Christina. Friendly robot for education and healthcare: fred. In: PROCEEDINGS of the 2023 ACM International Conference on Interactive Media Experiences Workshops. [S.l.: s.n.], 2023. P. 19–22.
- ROSÉN, Julia; LINDBLOM, Jessica; LAMB, Maurice; BILLING, Erik. Previous experience matters: an in-person investigation of expectations in human–robot interaction. **International journal of social robotics**, Springer, v. 16, n. 3, p. 447–460, 2024.
- RUDENKO, Irina; NORMAN, Utku; MAURE, Romain; RUDENKO, Andrey; WEINBERGER, Nora; KREBS, Franziska; PELLER-KONRAD, Fabian; REISTER, Fabian; ASFOUR, Tamim; BRUNO, Barbara. Drawings for insight on preschoolers’ perception of robots. In: COMPANION of the 2024 ACM/IEEE International Conference on Human-Robot Interaction. [S.l.: s.n.], 2024. P. 920–924.
- SAMPAIO, Emmanuel VB; LÉVÊQUE, Lucie; SILVA, Matthieu Perreira da; LE CALLET, Patrick. Are facial expression recognition algorithms reliable in the context of interactive media? a new metric to analyse their performance. In: EMOTIONIMX: Considering Emotions in Multimedia Experience (ACM IMX 2022 Workshop). [S.l.: s.n.], 2022.
- SCHULZ, Trenton; HOLTHAUS, Patrick; AMIRABDOLLAHIAN, Farshid; KOAY, Kheng Lee; TORRESEN, Jim; HERSTAD, Jo. Differences of human perceptions of a robot moving using linear or slow in, slow out velocity profiles when performing a cleaning task. In: IEEE. 2019 28th IEEE international conference on robot and human interactive communication (RO-MAN). [S.l.: s.n.], 2019. P. 1–8.
- SERENGIL, Sefik Ilkin; OZPINAR, Alper. Hyperextended lightface: a facial attribute analysis framework. In: IEEE. 2021 International Conference on Engineering and Emerging Technologies (ICEET). [S.l.: s.n.], 2021. P. 1–4. DOI: [10.1109/ICEET53442.2021.9659697](https://doi.org/10.1109/ICEET53442.2021.9659697).
- SHAMSUDDIN, Syamimi; YUSSOF, Hanafiah; ISMAIL, Luthffi Idzhar; MOHAMED, Salina; HANAPIAH, Fazah Akhtar; ZAHARI, Nur Ismarrubie. Initial response in hri-a case study on evaluation of child with autism spectrum disorders

- interacting with a humanoid robot nao. **Procedia engineering**, Elsevier, v. 41, p. 1448–1455, 2012.
- SHANNON, Claude Elwood. A mathematical theory of communication. **The bell system technical journal**, Musikalienhandel, v. 27, n. 3, p. 379–423, 1948.
- SHIBATA, Takanori. An overview of human interactive robots for psychological enrichment. **Proceedings of the ieee**, IEEE, v. 92, n. 11, p. 1749–1758, 2004.
- _____. Therapeutic seal robot as biofeedback medical device: qualitative and quantitative evaluations of robot therapy in dementia care. **Proceedings of the ieee**, IEEE, v. 100, n. 8, p. 2527–2538, 2012.
- SHIBATA, Takanori; HUNG, Lillian; PETERSEN, Sandra; DARLING, Kate; INOUE, Kaoru; MARTYN, Katharine; HORI, Yoko; LANE, Geoffrey; PARK, Davis; MIZOGUCHI, Ruth, et al. Paro as a biofeedback medical device for mental health in the covid-19 era. **Sustainability**, MDPI, v. 13, n. 20, p. 11502, 2021.
- SHIBATA, Takanori; INOUE, Kazuyoshi; IRIE, Robert. Emotional robot for intelligent system-artificial emotional creature project. In: IEEE. PROCEEDINGS 5th IEEE International Workshop on Robot and Human Communication. RO-MAN'96 TSUKUBA. [S.l.: s.n.], 1996. P. 466–471.
- SICILIANO, Bruno; KHATIB, Oussama. Robotics and the handbook. In: SPRINGER Handbook of Robotics. [S.l.]: Springer, 2016. P. 1–6.
- SOUZA, Tainá G; FLORES, Marcia; MONTEIRO, Nathália; ELLER, Estéfani; CABRAL, Sandy; LIMA, Adelmo; NASCIMENTO, Aline P; MUCHALUAT-SAADE, Débora C; SALGADO, Luciana CC. Hackathon 360: estimulando meninas na computação em projetos de robótica social para educação. In: SBC. WOMEN in Information Technology (WIT). [S.l.: s.n.], 2025. P. 139–150.
- SPITALE, Micol; AXELSSON, Minja; JEONG, Sooyeon; TUTTÖSI, Paige; STAMATIS, Caitlin A; LABAN, Guy; LIM, Angelica; GUNES, Hatice. Past, present, and future: a survey of the evolution of affective robotics for well-being. **Ieee transactions on affective computing**, IEEE, 2025.
- SPITALE, Micol; GUNES, Hatice. Affective robotics for wellbeing: a scoping review. In: IEEE. 2022 10th International conference on affective computing and intelligent interaction workshops and demos (ACIIW). [S.l.: s.n.], 2022. P. 1–8.

- SUBEDI, Anushka; PANDEY, Dipesh; MISHRA, Deepti. Programming nao as an educational agent: a comparison between choregraphe and python sdk. In: SPRINGER. THE Proceedings of the International Conference on Smart City Applications. [S.l.: s.n.], 2021. P. 367–377.
- SZÜCS, Veronika; KÁROLYI, György; TATÁR, András; MAGYAR, Attila. Voice controlled humanoid robot based movement rehabilitation framework. In: 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). [S.l.: s.n.], 2018. P. 000191–000196. DOI: [10.1109/CogInfoCom.2018.8639704](https://doi.org/10.1109/CogInfoCom.2018.8639704).
- TÄRNING, Betty; TJØSTHEIM, Trond A; JOHANSSON, Birger. Communicating emotional state and personality with eye-color and light intensity. In: PROCEEDINGS of the 7th International Conference on Human-Agent Interaction. [S.l.: s.n.], 2019. P. 214–216.
- TATSUMI, Akihiro; OKUBO, Masashi. Influences of changing clothes color of communication robot on information transmission to listener. **Journal of advanced mechanical design, systems, and manufacturing**, The Japan Society of Mechanical Engineers, v. 18, n. 3, jamdsm0031–jamsdm0031, 2024.
- THALMANN, Nadia Magnenat. Social robots: their history and what they can do for us. **Perspectives on digital humanism**, Springer International Publishing, p. 9–17, 2022.
- TSVETKOVA, Paulina; MITEVSKA, Mayiana. Supporting emotional well-being and social skills through social robots. **International journal of management studies and social science research (ijmsssr)**, IJMSSSR, v. 7, n. 1, p. 315–322, 2025. ISSN 2582-0265. Available from: <https://www.ijmsssr.org/paper/68c914768d355.pdf>.
- TUTUL, R; BUCHEM, I; JAKOB, A. Towards a gamified human-robot interaction framework for enhanced learning. In: IATED. ICERI2024 Proceedings. [S.l.: s.n.], 2024. P. 5412–5421.
- ULUER, Pinar; KOSE, Hatice; GUMUSLU, Elif; BARKANA, Duygun Erol. Experience with an affective robot assistant for children with hearing disabilities. **International journal of social robotics**, Springer, v. 15, n. 4, p. 643–660, 2023.
- ULUER, Pinar; KOSE, Hatice; OZ, Bulent Koray; AYDINALEV, Turgut Can; BARKANA, Duygun Erol. Towards an affective robot companion for audiology rehabilitation: how does pepper feel today? In: IEEE. 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). [S.l.: s.n.], 2020. P. 567–572.

- VAHRENKAMP, Nikolaus; WÄCHTER, Mirko; KRÖHNERT, Manfred; WELKE, Kai; ASFOUR, Tamim. The robot software framework armarx. **It-information technology**, De Gruyter Oldenbourg, v. 57, n. 2, p. 99–111, 2015.
- VASHISHTH, Tarun Kumar; SHARMA, Vikas; SHARMA, Kewal Krishan; SHARMA, Rajeev; SHARMA, Mukesh Kumar. Affective computing in social robotics. In: **HUMANIZING Technology With Emotional Intelligence**. [S.l.]: IGI Global Scientific Publishing, 2025. P. 361–384.
- VASYLKIV, Yurii; MA, Zhen; LI, Guangliang; BROCK, Heike; NAKAMURA, Keisuke; POURANG, Irani; GOMEZV, Randy. Shaping affective robot haru's reactive response. In: **IEEE. 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)**. [S.l.: s.n.], 2021. P. 989–996.
- VINAY, Rasita; TOMMILA, Nora Camilla; SCHLÖGL, Mathias; KLÖPPEL, Stefan; BILLER-ANDORNO, Nikola; KOWATSCH, Tobias. Grace: towards an embodied voice assistant for improving quality of life by leveraging elements of cognitive stimulation therapy. In: **PROCEEDINGS of Mensch und Computer 2024**. [S.l.: s.n.], 2024. P. 560–565.
- VINAY, Rasita; UETOVA, Ekaterina; TOMMILA, Nora Camilla; BILLER-ANDORNO, Nikola; KOWATSCH, Tobias. A hybrid rule-and large language model-based embodied voice assistant (grace) for cognitive stimulation in older adults: usability study assessing technical feasibility, technology acceptance, and working alliance. **Jmir aging**, JMIR Publications Inc., Toronto, Canada, v. 8, n. 1, e76489, 2025.
- VITALE, Antonio; IACONO, Umberto Dello. Using social robots as inclusive educational technology for mathematics learning through storytelling. **European public & social innovation review**, v. 9, p. 1–17, 2024.
- WALLISER, Florianne; BILLER-ANDORNO, Nikola; KOWATSCH, Tobias; VINAY, Rasita. Evaluating the grace voice assistant for dementia care among caregivers and healthcare professionals: an interview study. **Digital health**, SAGE Publications Sage UK: London, England, v. 12, p. 20552076261416341, 2026.
- WANG, Kai; SANG, Guo-Yuan; HUANG, Lan-Zi; LI, Shi-Hua; GUO, Jian-Wen. The effectiveness of educational robots in improving learning outcomes: a meta-analysis. **Sustainability**, MDPI, v. 15, n. 5, p. 4637, 2023.

- WILLIAMS, A Jess; FREED, Maureen; THEOFANOPOULOU, Nikki; DAUDÉN ROQUET, Claudia; KLASNJA, Predrag; GROSS, James; SCHLEIDER, Jessica; SLOVAK, Petr. Feasibility, perceived impact, and acceptability of a socially assistive robot to support emotion regulation with highly anxious university students: mixed methods open trial. **Jmir mental health**, JMIR Publications Toronto, Canada, v. 10, e46826, 2023.
- WOOLF, Bobby. Null object. In: MARTIN, Robert C.; RIEHLE, Dirk; BUSCHMANN, Frank (Eds.). **Pattern languages of program design 3**. [S.l.]: Addison-Wesley, 1998. P. 5–18.
- XU, Junchao; BROEKENS, Joost; HINDRIKS, Koen; NEERINCX, Mark A. **Mood expression through parameterized functional behavior of robots**. [S.l.]: IEEE, 2013.
- YI, Seung-Joon. Software framework for an intelligent mobile manipulation robot. In: IEEE. 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT). [S.l.: s.n.], 2018. P. 1–4.
- ZHAO, Wenzheng; GANGARAJU, Kruthika; YUAN, Fengpei. Multimodal perception-driven decision-making for human-robot interaction: a survey. **Frontiers in robotics and ai**, Frontiers Media SA, v. 12, p. 1604472, 2025.
- ZHOU, Shujie; TIAN, Leimin. Would you help a sad robot? influence of robots' emotional expressions on human-multi-robot collaboration. In: IEEE. 2020 29th IEEE international conference on robot and human interactive communication (RO-MAN). [S.l.: s.n.], 2020. P. 1243–1250.

APPENDIX A - BACKGROUND

This appendix introduces some important concepts for understanding both the objects that are at the center of this thesis’s research, social robots and affective robotics, and some of the contributions presented as extensions of existing elements. It covers the concept of social robots and their applications in healthcare and education, as well as the concept of affective robotics, which aims to give the robots with the ability to recognize and effectively express user emotions, potentially enhancing user engagement during interactions. Furthermore, the social robot EVA, its programming language EvaML, and its simulator EvaSIM are introduced in their original versions. The chapter is organized as follows: Section [A.1](#) introduces social robots; the concept of affective computing is discussed in Section [A.2](#); and the EVA robot, the EvaML language, and the EvaSIM simulator are presented in sections [A.3](#), [A.4](#), and [A.5](#), respectively.

A.1 Social Robots

Social robots, unlike industrial robots, are specifically designed to interact with human beings and are not treated merely as technological devices but as social agents ([FERRARI; EYSSEL, 2016](#)). Their interactions utilize natural communication channels, such as voice, gestures, body movements, and touch. Their ability to operate in accordance with established social and cultural norms is a key characteristic ([LI; JOHN-JOHN; TAN, 2011](#)). Social robots often feature anthropomorphic or zoomorphic forms, which facilitate emotional acceptance and the establishment of rapport with the user ([MUÑOZ, 2020](#); [MOKHTAR, 2019](#)).

The primary function of social robots is to foster positive outcomes in human interactions, and they can operate in various fields such as education and healthcare ([MUÑOZ, 2020](#)). They can serve as valuable complements to traditional therapeutic and educational practices by promoting emotional well-being and the development of social skills ([TSVETKOVA; MITEVSKA, 2025](#)).



Figure 68: Seal robot, Paro, interacting with lady with dementia (left) and in a pediatric intensive care unit interacting with an 11-year-old boy (right) (SHIBATA; HUNG, et al., 2021).

One of the first social robots applied to healthcare cited in the literature, the PARO robot (*Seal Robot*), was designed specifically for therapeutic purposes. Research and development of Paro started in 1993 (SHIBATA; INOUE; IRIE, 1996). The robot has a set of sensory capabilities, such as light sensors, touch sensors under its artificial skin, voice recognition microphones, temperature sensors, to control its body temperature and posture sensors to help knowing which position it is being held. It meets the perceived growth in the needs of older adults, including those requiring long-term care, to receive meaningful and positive social and emotional stimulation. At the same time, the use of the robot gives caregivers more freedom to perform other tasks (BISCHOFF; GRAEFE, 2002). Figure 68 shows an image of the robot interacting with a lady with dementia (on the left) and the robot interacting with an 11-year-old boy in a pediatric intensive care unit (on the right).

The NAO robot is perhaps the most widely used robot in literature. It is a humanoid robot developed by Alderbaran Robotics. It weighs around 4.5kg and is 57cm tall. It has 21 degrees of freedom (DOF). In addition to its ability to move, as it is a robot manufactured by a company, several human-robot interaction studies use NAO, whether in applications in the area of healthcare (SHAMSUDDIN et al., 2012), (NEERINCX et al., 2023) or in education (LOPEZ-CAUDANA et al., 2022), (NAMLISESLI et al., 2024), (FRAGAKIS; TRICHOPOULOS; CARIDAKIS, 2025). Despite its remarkable ability to move, pose, and express itself through body gestures (ERDEN, 2013), NAO is quite limited when it comes to expressing affection, as its ability to generate facial expressions is practically nonexistent, due to the design of its face. Figure 69 shows an image of NAO



Figure 69: Image of the NAO robot (left). Child with ASD interacting with NAO ([SHAM-SUDDIN et al., 2012](#)).

(on the left) and an image of a child with Autism Spectrum Disorder (ASD) interacting with the robot (on the right).

Another robot widely used in literature is the Pepper robot. Like NAO, Pepper is a robot developed by a private company, Softbank Robotics Corp. Pepper does not have as many degrees of freedom (DOF) as NAO, and also has the same limitations for generating facial expressions due to the design of its face. As a difference in relation to NAO, Pepper has a tablet on its chest, allowing users to interact with the robot via touch screen. [Uluer, Kose, Gumuslu, et al. \(2023\)](#) used Pepper in a system developed for audiometry testing and rehabilitation of children in a clinical environment. Figure 70 contains an image of two children with hearing disabilities playing the auditory perception game with Pepper. [Faisal et al. \(2024\)](#) proposed a humanoid robot-based solution to help Alzheimer's patients refresh their memory using Pepper. Aiming to improve learning outcomes in educational environments, [Tutul, Buchem, and Jakob \(2024\)](#) used the Pepper robot to define a framework for integrating gamification principles into human-robot interaction (HRI).

[Levinson et al. \(2021\)](#) evaluated the effectiveness and feasibility of integrating social robots into educational activities, particularly in non-STEM (Science, Technology, Engineering, and Mathematics) contexts. The robots were used to teach Hebrew language morphology to children. The authors compare the positive educational outcomes promoted by the robots in children during learning activities. The work uses the NAO robot, which is a proprietary and expensive robotics platform, together with the Patrice robot, which is a low-cost, 3D-printed robot with a puppet-like appearance. The quanti-



Figure 70: Children with hearing disabilities playing the auditory perception game with the robot Pepper ([ULUER; KOSE; GUMUSLU, et al., 2023](#)).

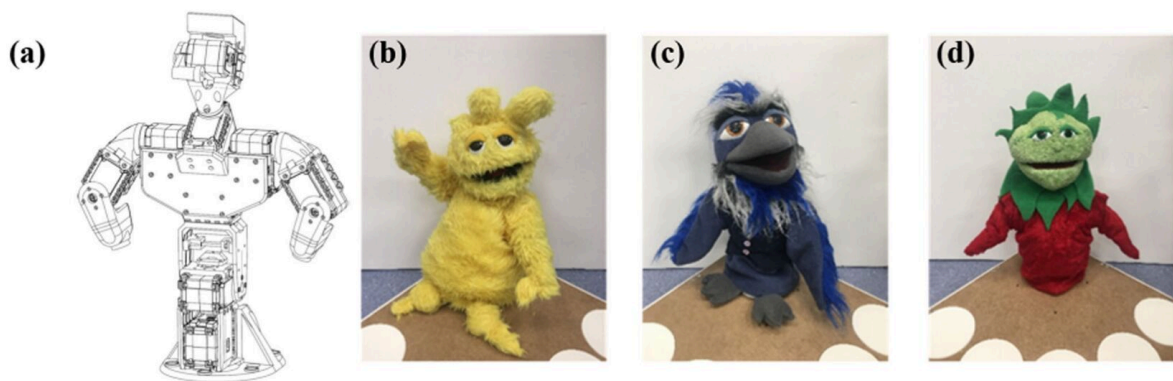


Figure 71: Patricc robotic platform. Mechanical structure (a). Puppet costumes (b-d) ([LEVINSON et al., 2021](#)).

tative results obtained from the use of the robots showed that children showed significant improvements in word memorization exercises after activities with the robots, indicating that both Patricc (a low-cost robot) and NAO were effective in promoting learning. The Patricc robot was designed to be a highly interactive solution for educational environments, allowing an affordable and effective approach to teaching and research in human-robot interaction. Figure 71 shows the mechanical structure of the Patricc robot with its puppet costumes.

The TINY (Teach, Inspire, Nurture, Yield) educational robot was proposed by [Jobaida et al. \(2024\)](#). The device was developed with the aim of improving children's learning (children aged 3 to 11), including children with learning disabilities, by helping them develop literacy and reasoning skills and apply simple mathematical concepts. The TINY robot, which can be seen in Figure 72, is built with low-cost components, such as the Rasp-

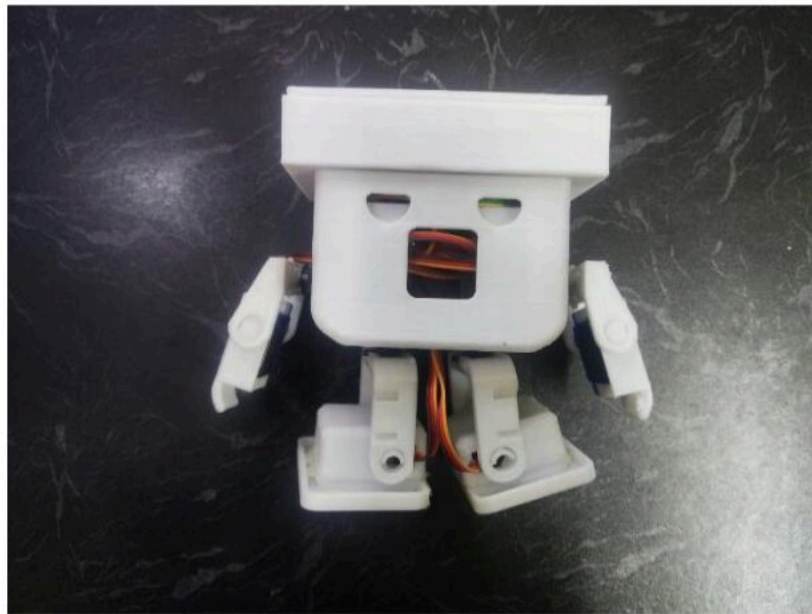


Figure 72: Robot TINY (Teach, Inspire, Nurture, Yield) ([JOBALDA et al., 2024](#)).

berry Pi 3 board, which acts as the main processing unit. In addition, TINY includes sensors (infrared, touch, temperature sensors), servomotors for movement, a microphone for voice commands, and a speaker for audible feedback. Its body is 3D printed, making it lightweight and resistant. Speech recognition is performed through cloud platforms such as Google Speech-to-Text, while dialogue management is done with tools such as Dialogflow, allowing TINY to respond appropriately to children's questions.

A.2 Affective Robotics

Affective robotics is an important area of human-robot interaction (HRI) research, investigating how robots can express and recognize emotions to improve their acceptance and functionality in social environments. Developing robots capable of expressing affection through different modalities is essential to make them more effective in domains such as healthcare, education, and personal assistance. Several studies have been conducted evaluating the effectiveness of different approaches to emotional communication in robots, exploring modalities such as body movements, facial expressions, non-linguistic sounds and colors.

The use of affective robots in educational applications has been shown to improve learning outcomes. In ([WANG et al., 2023](#)), a meta-analysis of educational robot interventions found a moderate but significantly positive effect on learning outcomes with

robot use.

Affective robots contribute to improving emotional well-being in both educational and healthcare environments. They can provide emotional support, reduce anxiety and stress, and promote positive emotions, as demonstrated by studies involving robots such as NAO and Purrble ([NAMLISESLI et al., 2024](#); [WILLIAMS et al., 2023](#)). They are capable of providing personalized interactions through recognition and response to human emotions, these robots can adapt their interactions to meet the specific needs of individuals, whether in a classroom or in a therapeutic environment ([VASHISHTH et al., 2025](#); [MISHRA; TRIPATHI; KUMAR, 2024](#)).

[Cárdenas et al. \(2024\)](#) used the Pepper robot to express emotions by combining movements and emojis (which were displayed on the tablet on its chest), revealing that the combination of visual cues makes the robot's effective expression clearer. In addition to facial expressions, body movements play an important role in affective communication. The study carried out by [Beck et al. \(2013\)](#) reinforces this idea by analyzing how children interpret the body language of robots, indicating that subtle adjustments, such as head tilt, can alter the emotion perceived by the user.

The use of non-linguistic sounds (NLUs) is another approach to robot emotional communication. [Belpaeme et al. \(2018\)](#) explore how different types of sounds are perceived by users, showing that high-pitched and short sounds are associated with positive emotions, while low-pitched and prolonged sounds are associated with negative emotions. When combined with other elements, such as facial expressions and movements, these sounds can significantly improve the clarity of the robot's affective expressions. Voice modulation also influences emotional perception in human-robot interactions. In ([LI; ISHI, et al., 2017](#)), a system based on prosody analysis and the sentiment contained in the user's speech text was implemented to generate more natural emotional responses in the robot. The results showed that the combination of voice tone and textual content improves the emotional response.

Using color to represent emotions is a common strategy in affective robotics. In ([TATSUMI; OKUBO, 2024](#)), a robot with translucent clothing and internal lighting was tested to see if color changes helped in the expression of emotions. Although the colors were effective when participants understood their logic, in some cases the emotions expressed were not clearly identified by users. Another study, ([TÄRNING; TJØSTHEIM; JOHANSSON, 2019](#)), analyzed how varying the color of a robot's eyes can influence the user's perception of its emotion and personality. The results suggest that vibrant colors

are more effective in communicating positive emotions, while darker tones are perceived as more representative of negative emotions.

A.3 EVA Robot

The EVA (*Embodied Voice Assistant*) robot is a social robot and was originally developed by researchers at Center for Scientific Research and Higher Education of Ensenada (CI-CESE) in Baja California, Mexico. This section presents the original version of the robot, which was developed as part of a doctoral thesis by [Cruz Sandoval \(2020\)](#), with extensions related to the use of the USB webcam, a smart bulb, and facial emotion recognition that were added in the work of (**towardsEVA**).

A.3.1 Robot Architecture

A.3.1.1 Hardware Components

Regarding hardware components, Figure 73 shows the basic version of the EVA robot and its main components. This version of the robot uses a Raspberry Pi 4 as its core component and includes a voice interface, a touchscreen, an RGB LED ring integrated into the chest, and servomotors that allow the head to move with two degrees of freedom. The new elements, a webcam and a smart bulb, added to the robot in (**towardsEVA**) can also be seen in the figure.

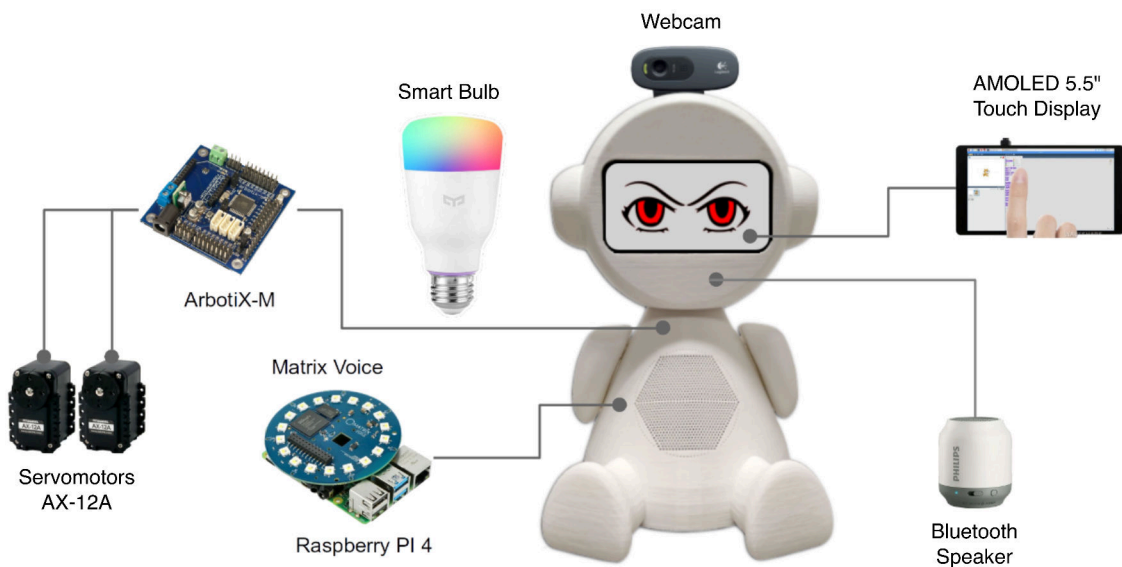


Figure 73: EVA Robot Hardware Components.

As previously mentioned, the EVA robot was created by researchers at CICESE in Mexico. Several studies using these and other intermediate versions of the robot (even simpler prototypes) have been published. [Cruz-Sandoval and Favela \(2019\)](#) present the results of a study using the robot to conduct a non-pharmacological therapy session with elderly patients with dementia.

A.3.1.2 Software Components

This Mexican version of EVA features a web application for script development. It uses a database to save and load created scripts and maintains a file system folder to store audio files that can be played during interactions. The web system was developed using the MEAN technology stack, which includes MongoDB, Express, AngularJS, and NodeJS. C++ was used to develop the firmware controlling the robot's servomotors, running on the ArbotiX-M board, and the code controlling the RGB LED animations, executed by the Matrix Voice board. Figure 74 illustrates the robot's software components.

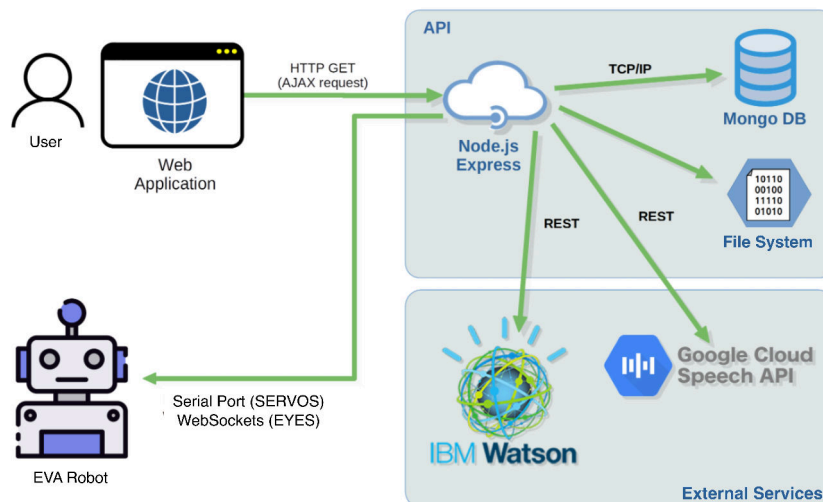


Figure 74: Software architecture of the EVA robot.

The EVA platform includes a component for developing interactive sessions. The user can create interaction scripts using a Visual Programming Language (VPL), defining the script flow using sequences, conditions, and loops ([MITJANS, 2020](#)). VPL has several elements that can be used to create interaction scripts, some of which are: an element for defining the language to be used by the speech-to-text and text-to-speech components, an element for controlling the robot's gaze expression, another component for voice recognition, a component for recognizing the user's facial expressions using the webcam, and a component capable of controlling sensory light effects using the smart bulb.

Figure 75 shows an interaction that was built using VPL. As a visual programming language, VPL does not have syntactic details like a conventional programming language. It was created with the aim of facilitating the construction of interaction scripts by people who are not programming experts. Using this graphical tool, built with the GoJS web framework¹, a script can be developed in a simple way, just by dragging and dropping the language control elements into the graphical interface.

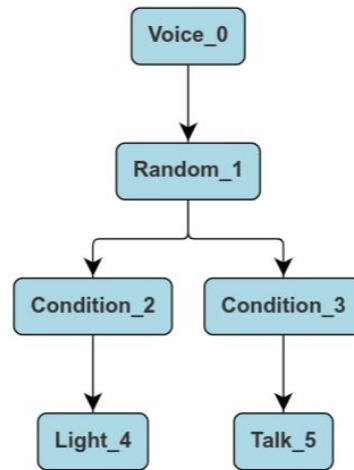


Figure 75: A script on VPL.

The execution flow of a script in VPL occurs from top to bottom and in the case of conditional elements, from left to right. A script in VPL is represented by a graph where the nodes are the elements of the language (robot capabilities) and the edges indicate the sequence of the execution flow. As can be seen in Figure 75, the script represented in the image has five distinct elements. The *Voice* element is used to configure the language and voice timbre used in the text-to-speech and speech-to-text processes. The *Random* element generates a random natural number within a range specified in its parameters. The *Condition* element evaluates a condition, in the example it evaluates the number randomly generated by the previous command and, depending on the result, the script execution flow can follow different paths. The element that follows the path on the left is *Light*, which controls the smart bulb, its state (on or off) and its color. The element that follows the path on the right is *Talk*, which makes EVA speak the text specified as a parameter. Each VPL element has a set of parameters that are configured when inserted into the script.

¹<https://gojs.net/latest/index.html>

A.4 EvaML

EvaML is a proposal of a DSL (Domain Specific Language) based on XML to create interactive sessions for the EVA robot, with the following goals:

1. facilitate the development of robot programs by people with technical knowledge in programming;
2. allow greater control of EVA commands and their respective parameters;
3. add abstractions of programming languages for developing scripts;
4. enable the development of scripts independent of the robot's control interface, that is, using any text editor.

All commands that control the robot's multimodal interaction elements are present in EvaML, including the *Light* component (which controls a smart bulb), the voice recognition commands and the *userEmotion* command that enables the recognition of the user's facial expression through a webcam. The language also has elements for creating and manipulating variables, generating random numbers, conditional controls using *switch* and *case* elements, among others. The EvaML parser automatically generates a corresponding script in the original JSON format that can be added to the robot's script database and then executed by the robot.

A.4.1 EvaML Document Sections

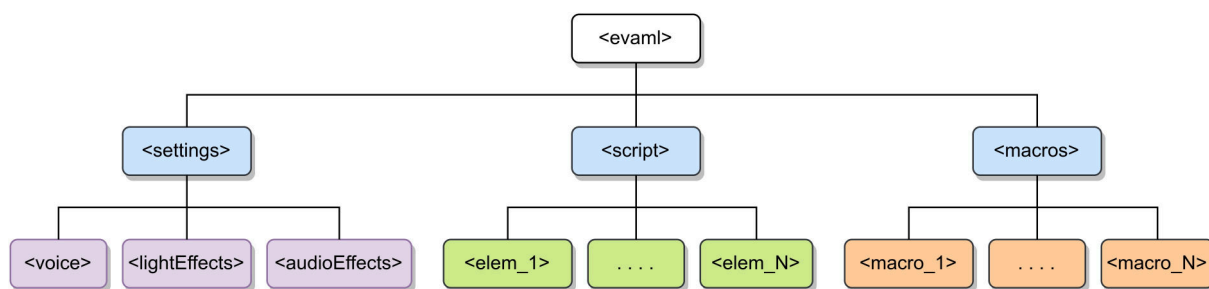


Figure 76: Tree structure of an EvaML document.

The tree structure of an EvaML document can be seen in Figure 76. Table 41 shows the root element of the EvaML document (`<evaml>`) and elements `<settings>`, `<script>` and `<macros>` that represent its main sections. The content and attributes of each element are presented. In the *Attribute* column, an underlined attribute indicates that it

Table 41: Content model of the main sections of an EvaML document.

Element	Attributes	Contents
evaml	<u>name</u>	(settings, script, macros?)
settings		(voice lightEffects? audioEffects?)
script		(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* motion* useMacro* listen* audio* led* counter* switch*)
macros		(macro+)

is required. In the *Content* column, symbols are used to indicate the order and number of times an element can or should occur. The character "," indicates that all listed child elements must be used in the given sequence. The character "|" indicates that any element can occur within the parent element. The character "+", on the other hand, indicates that the child element must appear one or more times. To indicate that the element can be used zero or more times within the parent element the character "*" is used. Finally, the character "?" indicates that the element is optional, that is, the element may not exist or there is only one occurrence of it.

Element `<settings>` - Defines some global characteristics of the script. It is possible to define the voice timbre and the language in which the robot will communicate. It is also possible to define whether the generated code, when executed, will reproduce commands for light effects, sound effects or even play music. By configuring these parameters, it is possible to globally modify the functioning of the script without having to directly change the definitions of its individual elements. Listing A.1 presents a code snippet that exemplifies elements `<evaml>` and `<settings>`.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <evaml name="script01">
3   <settings>
4     <voice tone="en-US_AllisonV3Voice" />
5     <lightEffects mode="ON" />
6     <audioEffects mode="ON"/>
7   </settings>
8   ...
9 </evaml>

```

Listing A.1: Example of the `<settings>` section.

Element `<script>` - Contains the sequence of commands that the robot must execute. You can see some of them in the code snippet presented in Listing A.2. In line 2, there is the `<light>` command, which turns on the smart bulb setting its color to blue. In the next line, the `<talk>` command makes the robot say something, in this example "Hi, how are you?". The `<wait>` command then causes the script to pause for 2000ms (2s). Next, the `<audio>` command plays an audio file called "song01". The `block` attribute set to "TRUE" makes the script continue running only after the audio file has finished playing. Then, the robot says "Bye!" and turns off the smart bulb.

```

1 <script>
2   <light state="ON" color="BLUE"/>
3   <talk>Hi, how are you?</talk>
4   <wait duration="2000" />
5   <audio source="song01" block="TRUE" />
6   <talk>Bye!</talk>
7   <light state="OFF" />
8 </script>

```

Listing A.2: Example of the `<script>` section.

Element `<macros>` - It is one of the abstractions created in the EvaML language, as can be seen in Listing A.3. It is possible to create subscripts as **macros** that can be referenced inside the `<script>` section to reuse them. A macro has the `id` attribute that serves to identify it. These macros can be used within the `<script>` section using the `<useMacro>` command. The `macro` attribute of the `<useMacro>` command references the `<macro>` element defined inside `<macros>`. During the parsing of the EvaML document, all macros are expanded with their content in the `<script>` section. There is no limit to the number of macros created, nor to the number of references to those macros in the script. As can be seen in Table 41, the `<macros>` element is not required.

```

1 <macros>
2   <macro id="GREETINGS">
3     <talk>Hello, I'm robot Eva</talk>
4     <talk>What is your name?</talk>
5     <listen />
6   </macro>
7 </macros>

```

Listing A.3: Example of the `<macros>` element.

A.4.2 EvaML Language Commands

The EvaML language offers commands that make it possible to interact with the robot's sensors and actuators. The language provides elements that allow the creation and manipulation of variables, as well as mathematical operations. It also presents elements for controlling the script's logic and execution flow. The robot's multimodal interaction elements can be easily controlled with EvaML, including elements for controlling light sensory effects and recognizing facial expressions.

Table 42 presents the content model of EvaML elements (commands) together with their attributes (optional and mandatory). To show more clearly the functionality of each EvaML command, Figure 77 shows an association between the language elements and the EVA robot functionalities. It is possible to see the commands associated with the robot's verbal and non-verbal communication capabilities. The `<light>` element, which controls the smart bulb, the `<userEmotion>` element, used in the facial expression recognition process, and the elements that work more directly with the robot's code execution unit, elements `<switch>`, `<case>`, `<counter>` and `<random>` that perform, respectively, logical, control, arithmetic and random number generation operations. The Listing A.4 shows a small example of the use of these language elements.

```

1 <script>
2   <counter var="SCORE" op="=" value="0" />
3   <random min="1" max="2" />
4   <switch var="$">
5     <case op="exact" value="1">
6       <light state="ON" color="GREEN" />
7       <counter var="SCORE" op="+" value="1" />
8     </case>
9     <case op="exact" value="2">
10      <evaEmotion emotion="HAPPY" />
11      <counter var="SCORE" op="+" value="2" />
12    </case>
13  </switch>
14 </script>

```

Listing A.4: Example of using the `<counter>`, `<random>`, `<switch>` and `<case>` elements.

Table 42: Content model of the EvaML elements (commands).

Element	Attributes	Contents
voice	<u>tone</u>	empty
random	id, <u>min</u> , <u>max</u> ,	empty
wait	id, <u>duration</u>	empty
talk	id	Text
stop		empty
light	id, <u>state</u> , color	empty
goto	<u>target</u>	empty
userEmotion	id	empty
evaEmotion	id, <u>emotion</u>	empty
motion	id, <u>type</u>	empty
useMacro	<u>macro</u>	empty
listen	id	empty
audio	id, <u>source</u> , <u>block</u>	empty
led	id, <u>animation</u>	empty
counter	id, <u>var</u> , <u>op</u> , <u>value</u>	empty
switch	<u>var</u>	(case*, default?)
macro	<u>id</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* macro* useMacro* listen* audio* led* counter* switch*)
case	<u>op</u> , <u>value</u>	(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* macro* useMacro* listen* audio* led* counter* switch*)
default		(random* wait* talk* stop* light* goto* userEmotion* evaEmotion* macro* useMacro* listen* audio* led* counter* switch*)

A.5 EvaSIM – EVA Simulator Software

EvaSIM is a software simulator for the EVA robot, aimed at serving as an auxiliary tool for the development of interactive sessions for EVA, or being used in the teaching/learning processes of the robot’s language and also as a tool for testing and debugging EVA scripts. The general architecture of EvaSIM can be divided into three main modules: the Parser Module, the Execution Module and the Graphical User Interface (GUI). An overview of this architecture can be seen in Figure 78.

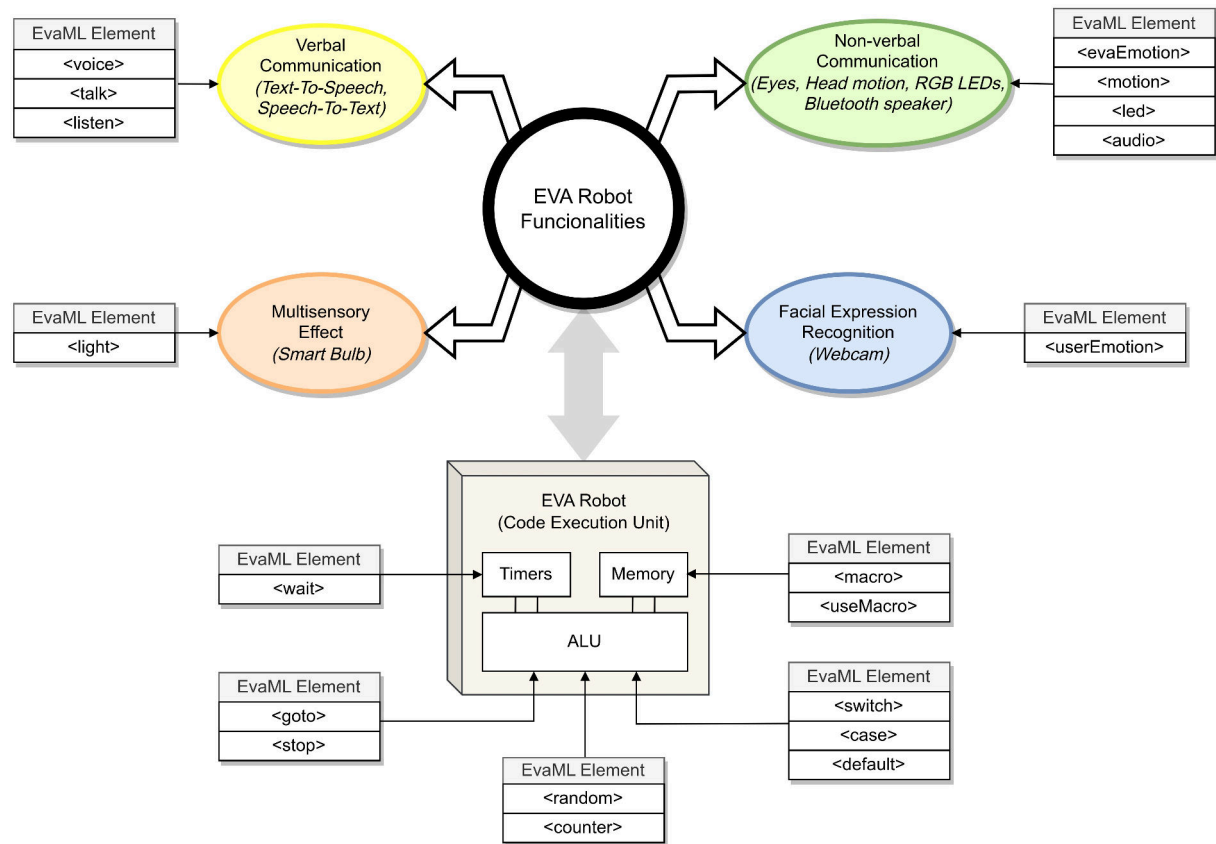


Figure 77: Association between EvaML language elements and EVA robot functionalities.

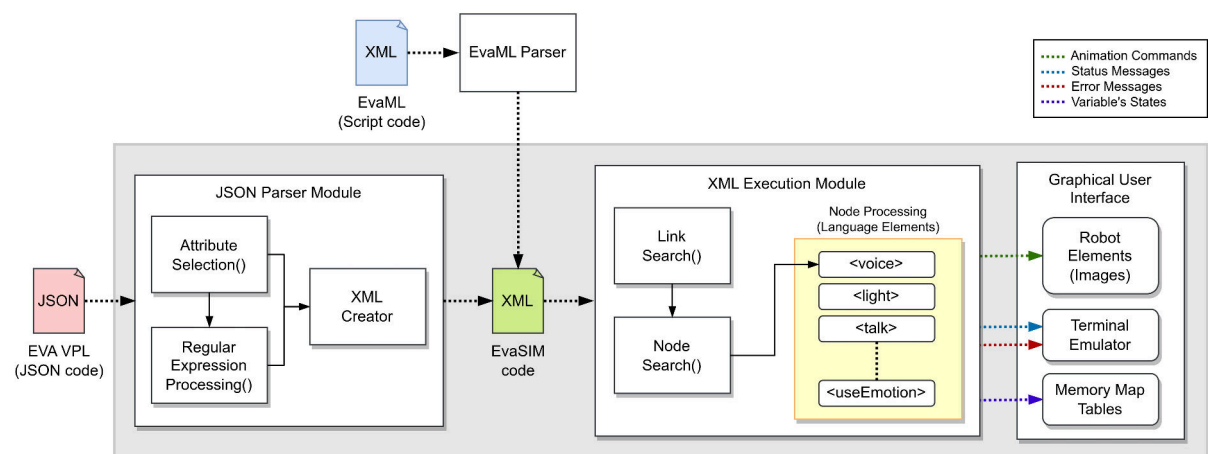


Figure 78: EvaSIM architecture overview.

A.5.1 The Parser Module

EvaSIM is capable of running a script created using the robot's visual programming language. For this, the simulator receives as input a JSON file that represents the graphic script, with its nodes, which represent commands of the robot language, and its edges, which are the connections between these nodes that define the script execution flow. As can be seen in Figure 79(a), a graphical script created in the VPL is stored in the robot's file system in textual format (JSON file). The JSON file represents the structure of nodes and edges (links). Each stored node takes with it, in addition to information about the commands and their respective parameters, information about the graphic element that is represented in the VPL interface. The first module, the parser, has the function of transforming this input into JSON in an intermediate XML file called EvaSIM code. The robot's JSON file contains, in addition to nodes and edges, information about the graphic script, which is not used by the simulator, but only by the physical robot.

The parsing process is performed through three main functions contained in the module: the attribute selection function, which is responsible for selecting the properties of the file structure that are of interest to the parser, also transcoding some values to suit the internal structure of the command processing in EvaSIM; the function for processing regular expressions, which is used to convert some formats that are more suitable for XML processing, and the function that physically generates the EvaSIM XML file that will be used as input for the next module. We can also see in Figure 79(b) an example of this intermediate XML file that resulted from mapping the JSON file referring to a VPL script. The structure of the XML file can be observed, which contains a section called "script" that is defined between the tags <script> where the nodes that represent the commands with their respective attributes are located. Each element has an identifier attribute "id". The "links" section contains the links that connect these nodes and define the execution flow of the script to be executed. EvaSIM is capable of directly executing codes in the EvaSIM XML code format without the need to go through the parser module.

A.5.2 The Execution Module

After parsing the source file (the JSON representation of a VPL script) and generating the EvaSIM XML file or after receiving an XML file generated by the EvaML parser as input, the Execution Module is called. The EvaSIM XML file is read and processed by three internal functions. The first one searches the links in the list of links and separates the nodes that are connected to them. These nodes are identified by their "id" attribute.

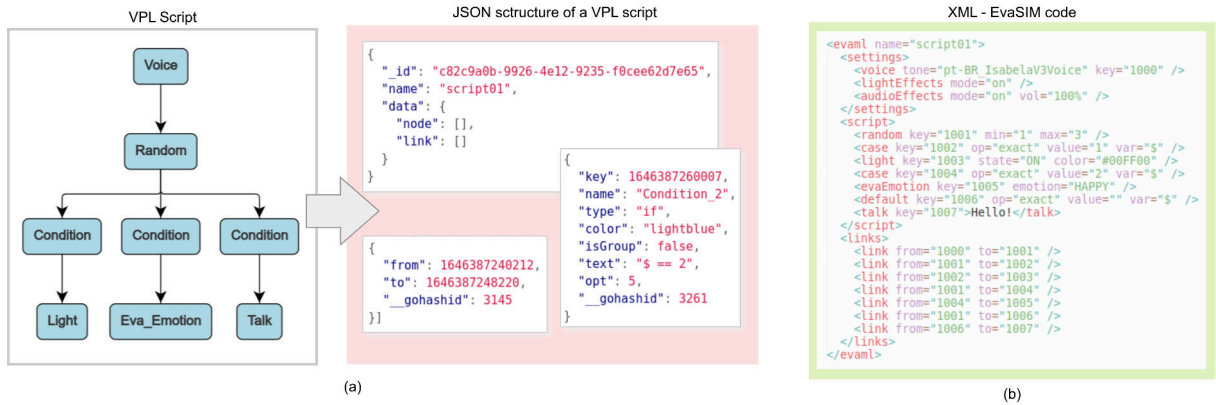


Figure 79: JSON structure of a VPL script (a) and an EvaSIM code representation example (b)

The node identifiers are then passed to the module's second function, which does the node search and selection. After being selected, the node is passed to the third function that effectively processes it based on its type and its corresponding attributes. This node processing function analyzes the node type and looks for it in the simulator's command list.

A command executed by the simulator can generate one of four signals: animation commands, status messages, error messages and variable states. These signals are sent to the third module of the architecture, the graphical user interface module. Signals of the "animation command" type cause some graphic animation to be executed in the simulator graphical interface, for example, the command that makes the robot express an emotion through the display. Signals of the "status message" type serve to indicate which kind of action is being performed by the simulator, such as, for example, the creation of a variable, a mathematical operation on it, a sentence being spoken by the simulator, etc. Signals of the "error messages" type serve to indicate an error in the code execution. The "states of variables" type signals carry with them the value contained in a variable that has just been referenced, either at the time of its creation or after an operation on it.

A.5.3 Graphical User Interface Module

The third simulator module is the GUI module that provides the graphical user interface. EvaSIM is a lightweight and easy to use application with three windows. Figure 80 shows the application's main window. This window can be divided into four distinct regions. The first region presents the graphic elements that represent the robot's functionalities. The elements indicated with gray circles (items 1 to 5) represent, respectively, the EVA

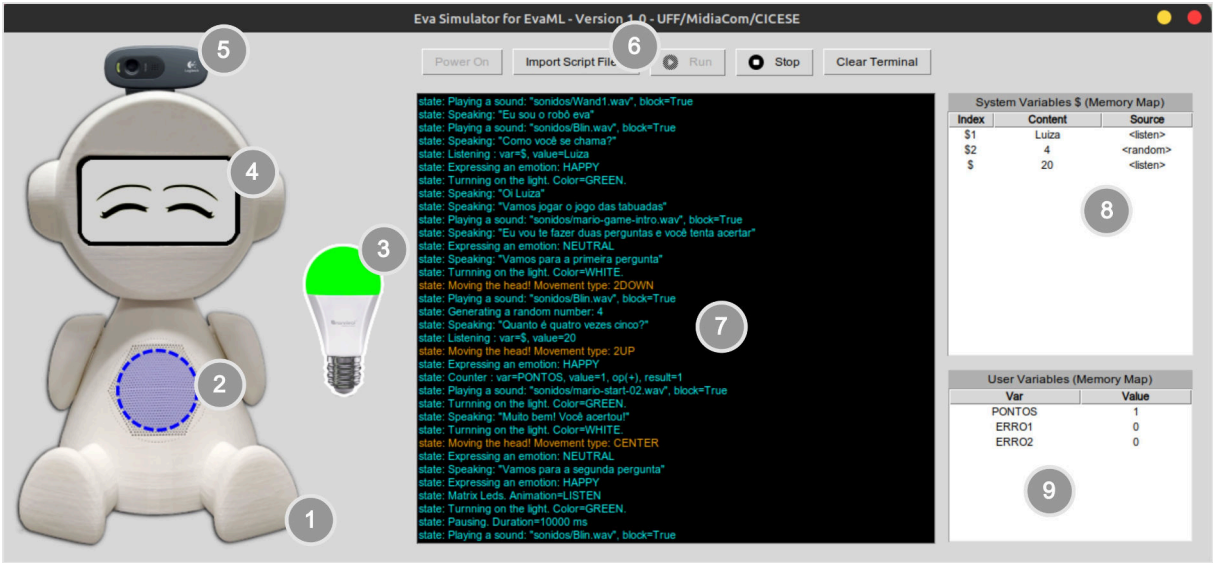


Figure 80: EvaSIM - Graphical User Interface elements.

body, the RGB LEDs on the robot's chest, the smart bulb, the robot's gaze and the webcam used in the facial expression recognition process. The second region (item 6) contains the button menu that controls the simulator. Using these buttons, you can import a script file, start and stop its execution, and also clean the terminal emulator. The third region (item 7) features a terminal emulator. The purpose of the emulator is to serve as a debugging tool in the execution of scripts. It displays information related to the creation and manipulation of variables, the texts that the robot must speak, the animations that are performed by the RGB LEDs, etc. The fourth and last region (items 8 and 9) presents two tables that show the status of the variables used in the script. The first table (item 8) presents the state of the variable that stores the values generated by the random number generation command, the values of the responses returned from the execution of the voice capture command and the results returned from the execution of element of facial expression recognition. This variable is automatically created during script execution. The second table (item 9) presents the state of the variables created by the script programmer.

As previously mentioned, EvaSIM uses two more windows. They are used to simulate the robot's voice recognition command and facial expression recognition command. These windows are shown in Figure 81. The window that simulates EVA's facial expression recognition command, the "userEmotion", presents 5 emotions in the form of *emojis*, allowing the user to select the desired emotion. The window that simulates the robot's voice recognition uses a text field where the user can enter the answer through the keyboard. The simulation of the movement of the robot's head is done by indicating

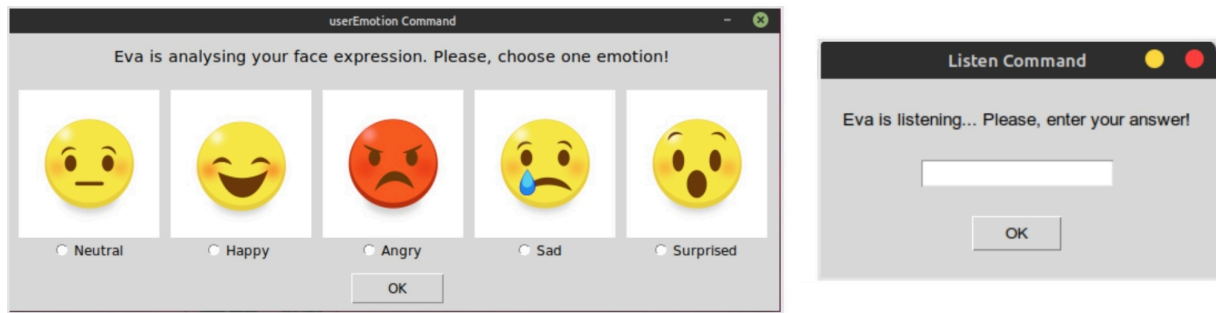


Figure 81: Facial expression capture (userEmotion) and voice capture (listen) simulation windows.

the movement (its type) being executed. This information is presented in text form in the simulator's terminal emulator.

This appendix introduced important concepts directly related to the contributions of this thesis, such as social robots (presenting key examples from the literature and their use in healthcare and education) and affective robotics (demonstrating how a robot's ability to express and recognize emotions can enhance its acceptance in social environments). Additionally, the EVA robot (in its original version), its script-based programming language (EvaML), and its simulator (EvaSIM) were introduced. The [Chapter 2](#) presents related work.

APPENDIX B - EVA ROBOT 2.0 - NEW 3D PARTS AND MODELS

In addition to the software extensions presented as part of this work, new 3D models of components for the EVA 2.0 robot are also proposed. Section [B.1](#) briefly describes the design of the PiCamera mount, and Section [B.2](#) discusses the design of the new movable arms. Finally, the model for the Arduino UNO R3 board mount is presented in Section [B.3](#).

B.1 3D Support for PiCamera

The original version of the EVA robot presented in ([ROCHA; VALENTIM, et al., 2022](#)), used a USB camera for image capture. This work proposes using a PiCamera instead of a USB camera. In addition to offering superior performance, since the PiCamera connects directly to the Raspberry Pi's dedicated camera slot—mounting, the camera on the robot's head allows it to change its field of view by moving the head. A new 3D-printed part is proposed to mount the new camera onto the robot's head.

Figure [82](#) shows the three items that make up the support that allows the PiCamera to be attached to the EVA's head without much difficulty. Item (1) in the figure serves to protect part of the camera and is screwed to item (2). Item (2) has a curved base that was modeled on the 3D model of the EVA's head. This item has two pins that serve for the initial fixation of the camera. Item (3) is used to close the support and fix the camera using two screws. Figure [83](#), in item (1), shows the image of the fully assembled part and in items (2) and (3) the profile and front images of the support already fixed to the EVA head. Figure [84](#) shows real images of the new 3D part assembled and attached to the EVA head.

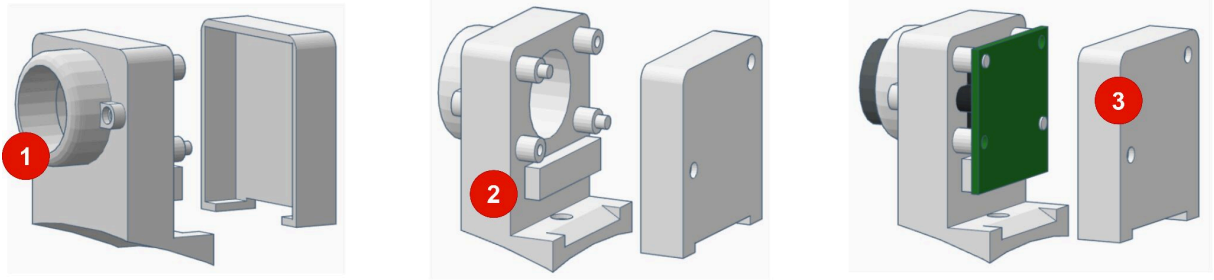


Figure 82: 3D elements that make up the support for attaching a PiCamera to the EVA 2.0 head.



Figure 83: Mounted support (1) and profile (2) and front (3) views of the support fixed to the EVA head.



Figure 84: Images of the camera support, its mounting on the robot's head and its connection to the Raspberry PI via the flat cable.

B.2 New Arms for EVA 2.0

In order to make the robot more expressive, this work presents a proposal for a new 3D model for the EVA arms that can move with the help of a servomotor. In order to add the ability to move the arms, it was first necessary to define what would be the best solution for the problem, since the arms had not been prepared for this purpose. The initial idea was to couple two servomotors to the internal structure of the EVA and then connect them to the robot arms through some type of axis. This approach presented a problem, since the robot arms move with a degree of inclination due to the shape of the robot's body. This issue made this first approach very complicated. A second idea would be to embed the servomotor inside the arm itself and then fix it to the robot body using some coupling part. This idea would eliminate the need for an axis coupled to the servomotor. To verify the feasibility of this new solution, two issues needed to be verified. First, it was verified, through the 3D models of the arm and servomotor, that the arm had a volume capable of holding the servomotor inside. Second, it was possible to verify that the original 3D printed robot arm was light and it would not be difficult for an SG90 micro servomotor to move it.

In order for the solution to be complete and to be implemented, in addition to the new arm model with the necessary cavity to receive the servomotor, it was necessary to create two more parts. The first of these would serve to connect the servomotor (already built into the new arm) to the EVA body, and the second part would be used to attach the first part to the robot body, but from the inside of the body.

Figure 85 shows the three 3D modeled parts and how they fit together. In item (3) we can see the new arm model with the space to receive the SG90 servomotor. This servomotor is screwed and fixed to the robot arm. Item (1) shows a detailed view of the part created to join the arm (with the servomotor already attached) to the robot body. As you can see, this part has, on its front part, small teeth that fit onto the servomotor shaft. This part is connected to the servomotor by means of a screw, as can be seen in item (4) of the same figure. Item (2) shows the second part created that serves to attach the first part to the robot body, but from the inside of the body.

In item (1) of Figure 86 you can see the arm attached to the body through part 1. In item (2), of the same figure, you can see the internal fixation of part 1 using the second designed part. Figure 87 shows real images of the 3D printed models. The 3D model files can be viewed and downloaded from this [link](#).

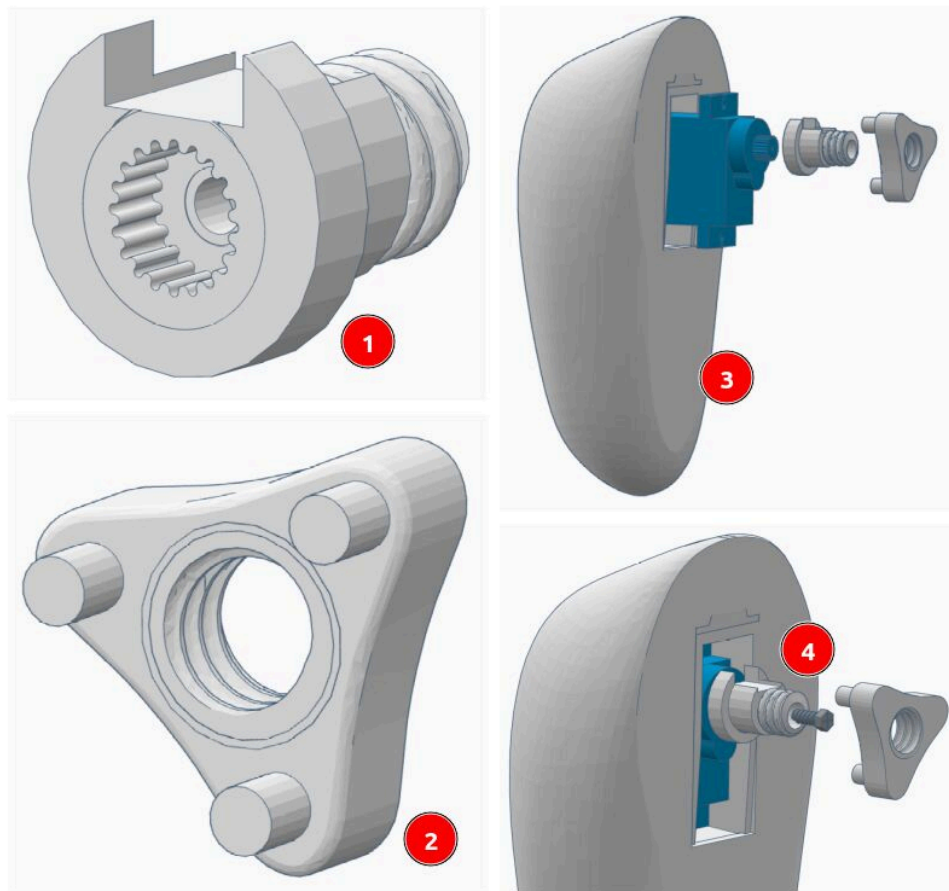


Figure 85: Main part that connects to the servomotor and the robot body (1). Part for internal fixing of the main part (2). New printed arm capable of holding the SG90 servomotor (3). View of the fittings of the 3 parts (4).

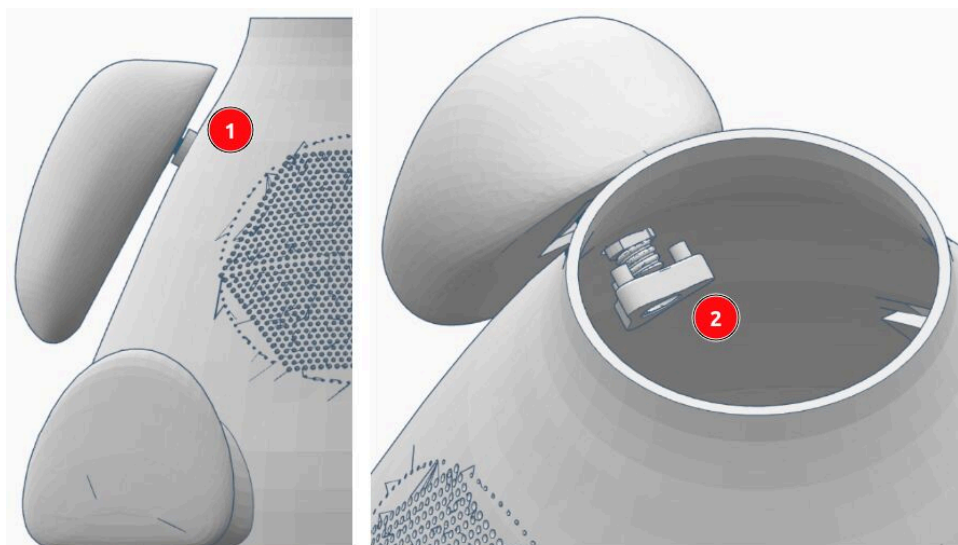


Figure 86: View of the fittings of the part to the arm (1) and the coupling of these parts to the body (2).

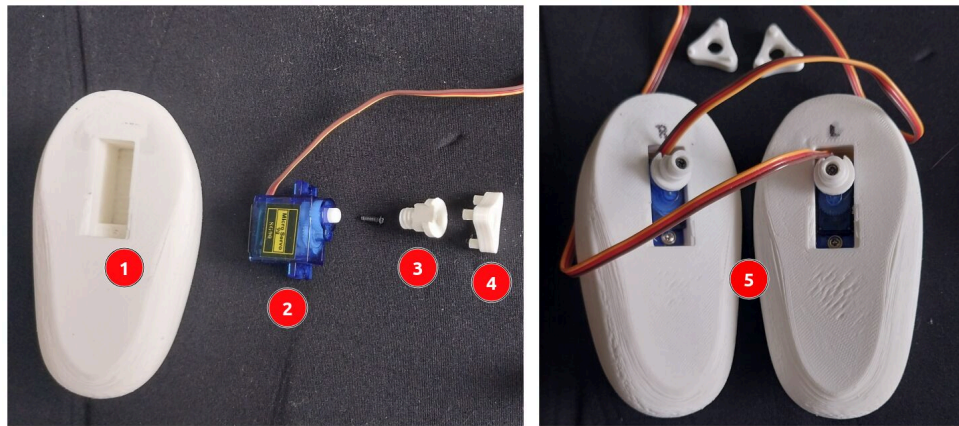


Figure 87: Real images of the created parts and the SG90 micro servomotor (1, 2, 3, 4). Two printed arms with embedded servomotors and connected parts (5).

B.3 Support for the Arduino UNO R3 Board

Initially, the software to control the EVA head movements was developed for an ArbotiX-M board. This board has the advantage of having inputs for direct connection of AX-12A servomotors, which are the motors used to move the robot's head. These dedicated connections allow, in addition to the logical connection with the servomotors, to supply them with a power source. This power source comes from an external source that can be connected directly to the board. Despite these positive points, the ArbotiX-M has been discontinued, making it difficult to obtain it and information regarding its hardware and software library. In addition, the only way to program this board, in an open-source manner, is through the Arduino IDE. However, due to the lack of updates, due to the discontinuation of the board's manufacture, only version 1.6 of the IDE is compatible with the available libraries.

The AX-12A servomotors, which are responsible for moving the robot's head, are "smart" servomotors and have a microcontroller inside them. The movement commands for the servomotor are sent via a serial communication established between the servomotor and the external control board.

In order to facilitate the construction of the EVA robot, with regard to controlling the robot's movements, in this new proposed version, the Arduino UNO R3 board was adopted as the control board for the servomotors. Arduino boards are cheaper and easier to find on the market. The AX12A.h library, available free of charge for Arduino, was used to program the AX-12A control software. This library allows the control of the AX-12A servomotors, allowing the definition of their operating mode (which can be in infinite motor mode or as a stepper motor), their position and speed.

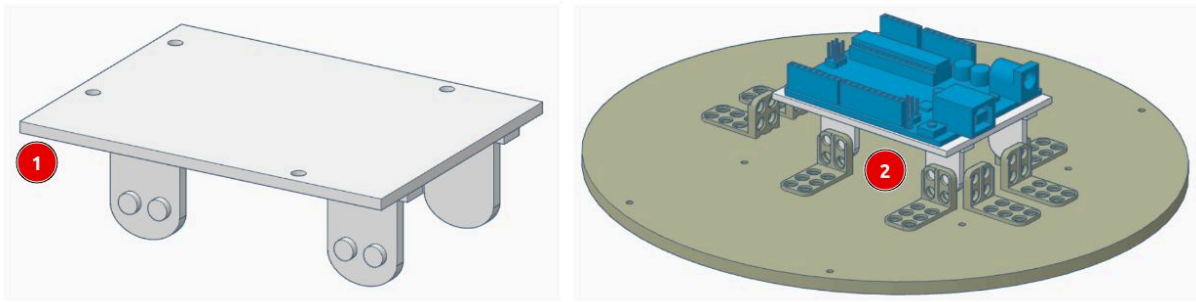


Figure 88: Support piece developed to accommodate the new control board, the Arduino UNO R3 (1). Coupling the support to the EVA base (2).

In order to integrate the Arduino board into the physical design of the robot, this paper proposes a model of a new 3D part. It can be fitted to the base of the robot body and allows the Arduino board to be screwed to it. Figure 88 shows the model of the new part (1) and its application to the base of the robot, supporting the Arduino board (2). The model file can be viewed and downloaded through this [link](#).

APPENDIX C - RESPONSES FROM THE FRAMEWORK EVALUATION CASE STUDIES

This appendix presents the questionnaire used in the framework evaluation conducted through research case studies, together with the complete responses provided by the participants. The questionnaire was designed to gather information regarding the use of the framework in real Human-Robot Interaction (HRI) projects, including aspects related to framework utilization, adaptation and extension, perceived benefits, encountered challenges, and suggestions for future improvements. To facilitate the presentation of the responses while preserving participant anonymity, the respondents are identified as P1, P2, and P3.

Evaluation of the Proposal for an Open Source Robotics Platform and Framework for the Development of Affective Social Robots

This set of questions aims to obtain insights from Human-Robot Interaction (HRI) researchers about the difficulties, challenges, criticisms, and suggestions related to the construction, assembly, control, and use of the provided framework.

The participants are identified throughout this appendix as P1, P2, and P3.

- P1 refers to the master's student who employed the framework in the development of the GRACE robot for cognitive intervention studies involving older adults.
- P2 refers to the master's student who employed the framework in the FRED MX project to investigate affective conversational strategies for supporting procrastination management in academic contexts.
- P3 refers to the undergraduate student who contributed to the technical development of the FRED MX platform, including robot assembly, hardware integration, connectivity troubleshooting, and the development of supporting software tools.

1) What is your name?

P1, P2 and P3

2) What is your institutional affiliation?

P1: ETH Zurich

P2: Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)

P3: Universidad Autónoma de Baja California

3) What is your research area in HRI (Human-Robot Interaction)?

P1: Interaction Design, Affective/Social Robotics, Ethics and Safety and Cognitive interventions

P2: Interaction Design

P3: Control Systems and AI

4) Please describe the application or experiment you implemented using the framework.

P1: We wanted to test different cognitive interventions delivered by a social assistive robot. We were able to develop and design four activities, including two cognitive interactions based on cognitive stimulation therapy. We developed a script that was partly rule based and partly leveraged generative AI. We did a first study in a controlled environment with healthy adults, and adjusted the script based on the feedback. We

then did a study in a controlled environment with older healthy adults, with the goal to later do a study with people with dementia.

P2: I used an existing affective conversational robot to explore support strategies for managing procrastination in an academic context. The work focused on applying and evaluating affective interaction strategies and conversational behaviors to support self-regulation and user engagement.

P3: I assembled the FRED robot for subsequent use at CICESE.

5) On a scale from 1 to 5 (1 = Very difficult, 5 = Very easy), how challenging was the learning curve to use the framework?

P1: Very easy (5)

P2: Neutral

P3: Easy (4)

6) What were the main difficulties you encountered when using the framework?

P1: As far as I remember, making scripts was very easy. What was more challenging was getting everything to work (both the physical robot as well as the software - I think I had some problems in the beginning with the virtual machines I was running the simulation software on, so I had to try out many different ones before it worked). Also adding new functionality was a bit more challenging, and we had some performance issues. I think the code was a bit challenging to understand and there was also limited documentation (in English) available.

P2: The main difficulties were understanding the overall structure of the framework and how the different modules and behaviors were interconnected, as well as adapting existing components to the specific goals of the study.

P3: FRED automatically disconnecting from the computer after a few minutes of inactivity (and sometimes spontaneous disconnection). On our FRED, there was also a connectivity issue where, if FRED made very abrupt movements, the internal connections to the power supply would disconnect.

7) On a scale from 1 to 5 (1 = No help, 5 = Excessive help), how much help did you need to use the framework?

P1: Moderate help (3)

P2: Moderate help (3)

P3: Moderate help (3)

8) Please describe the main difficulty you faced and the type of assistance you sought (e.g., documentation, community support, direct contact with the developer, etc.).

P1: I needed documentation and when that was not sufficient I sought direct contact with the developers. I think it was a bit difficult to understand who had developed what, which versions are open source, and who is the contact point for what. But in the end I received a lot of help which I am very grateful for! :)

P2: The main difficulty was understanding how to configure and adapt existing behaviors to the specific needs of the project. To address this, I mainly relied on the available documentation and direct guidance from the developers.

P3: The main problem was the interconnection between the ESP32 running LUA and the computer. To mitigate the problem, I researched the wireless communication protocol used (documentation) and contacted the developer for specific questions.

9) Did you need to extend or modify any of the existing "services" (modules, behaviors, etc.) in the framework to meet your project's needs? If so, which ones?

P1: Yes, we changed the facial expressions, the voice and added integration with ChatGPT.

P2: No major extensions were required. The work mainly involved configuring and adapting existing conversational and affective behaviors to fit the objectives of the study.

P3: Yes. Using Visual Studio, I developed a FRED control panel in C# to make it easier to control. I shared the link to the control panel with Marcelo in case he wanted to use it without any restrictions.

10) On a scale of 1 to 5 (1 = Very difficult, 5 = Very easy), how challenging was the task of extending or modifying these "services"? If you did not extend any functionality, select the "Not applicable (N/A)" option.

P1: Difficult (2)

P2: Not applicable (N/A)

P3: Very easy (4)

11) Did the framework meet your expectations regarding the design and implementation of behaviors for the robot? Why?

P1: As far as I remember, it was surprisingly easy to generate the scripts. The setup was the biggest challenge, since also some hardware parts were not available (the microphone was most challenging). What I think would have made things easier would be the possibility to change the Config from the EvaSIM directly (for instance the threshold for background sounds, since this was a big issue we faced and we often had to adjust it). Also the software structure was not the most intuitive to me.

P2: Yes, the framework met my expectations, as it provided a structured and flexible way to work with conversational and affective behaviors. This facilitated the exploration of socio-affective interaction strategies without requiring extensive low-level development.

P3: My education in Human-Robot Interaction is limited (I just finished my bachelor's degree), so I didn't have many expectations when starting this project. I must say, however, that I do see a lot of potential in the FRED project now that I understand its focus, its objectives, and the real impact it can have.

12) In what ways did the framework facilitate your research compared to alternative methods?

P1: The framework helped a lot:

- Creating the physical robot took little time

- Hardware components already tested in similar studies
- Having a software to use as a base and not having to create it all oneself Especially for a master thesis I would not have been able to do all of this myself.

P2: The framework facilitated the research by providing a structured environment for working with affective and conversational behaviors, reducing the need for low-level implementation. This allowed the focus to remain on interaction design and evaluation rather than on technical infrastructure.

P3: My focus was on manufacturing, and therefore this question does not apply to me.

13) What additional functionalities or "services" would you like the framework to offer to enhance your project?

P1: What I think would be useful is having a more modular software where one can easily plug different tools (change the voice recognition software, generative AI source etc.). Also the possibility to change the config from EvaSIM as mentioned earlier. Also having the possibility to start the script from any point would have been helpful.

P2: Additional hardware support such as an integrated small speaker for improved verbal interaction and a camera for facial affect recognition would enhance the framework. These components would allow richer multimodal interaction and more robust affective perception during human–robot interaction.

P3: I would request two things in particular:

1. Redesign the interior of FRED's shell to allocate a specific space for each cable and component. This would solve the component disconnection problem and make the robot sturdier so that children can use it.
2. Create a FRED Client running on a master device that can be reprogrammed to meet users' needs or to connect with other software.

14) Based on your experience, would you feel comfortable guiding another researcher or student to use the framework to build a similar affective robot to yours? Why?

P1: Partially. I think there were quite some difficulties with the hardware that I am not anymore sure how to solve (I think it took me 2 or 3 days to get the microphone to work). Else we had some difficulty with the robot sometimes stopping randomly or taking a very long time, but this could have also been to some of the changes I made. So I could help someone build a robot with the framework, but I am not sure if I could help them with all aspects.

P2: Yes, I would feel comfortable guiding others, as the framework provides clear structure and reusable components that support learning and experimentation, especially for students or researchers new to affective human–robot interaction.

P3: Yes, the framework is not complex to learn, and the results are solid despite the low cost. I think it is a good starting point for a project related to affective robots.

15) Beyond your application, what other areas or scenarios do you imagine the robot (hardware and software) or the framework could be used for?

P1: Maybe a learning assistant for children, a companion for lonely people (old & young), maybe in the waiting room of a psychologist to do some relaxation exercises with the patients before the session, ...

P2: The robot and framework could be applied in areas such as mental health support, healthcare assistance, social skills training, and user studies focused on long-term human–robot interaction.

P3: With more resources, FRED could be an extension into the physical world for an artificial intelligence model, like a virtual assistant similar to Alexa or Siri, or a learning assistant, connected to models like ChatGPT or Gemini.

16) Any other comments, suggestions, or feedback you would like to share?

P1: I am very grateful for all the help I got with the robot! :) Everyone was very kind and willing to help. I think it is a great project.

P2: Overall, the framework represents a valuable contribution to affective and social robotics research. Continued improvements in documentation and example scenarios would further support its adoption by the research community.

P3: I like that FRED is not trying to faithfully emulate the human form. I think there is more utility for a robot that adopts a more cartoonish aesthetic rather than an anthropomorphic one.