Rapport
Establishing Harmonious Relationships Between Robots and Humans
Bruno Henriques
Instituto Superior Técnico
Lisbon, Portugal
bruno.p.henriques@tecnico.ulisboa.pt

ABSTRACT
Autonomous agents are becoming our next companions. They may be able to offer physical therapy assistance, play games and even help us treat weight loss. However, it is not enough to build agents that create strong first impressions. They need to continuously convey such feelings and encourage user interactions, i.e., to build and maintain rapport over long periods of time. In order to manage rapport, agents need to show signs of positivity, mutual attention and coordination during, e.g., motivate, establish eye contact, and postural mimicry, respectively. Nowadays, social agents only tackle one of these components, and the ones that do are not robotic. For this purpose, we designed an extensible rapport model that enables robotic and virtual agents to show natural signs of rapport according to the dyadic state of the interaction. The model was implemented using the Socially Expressive Robotics Architecture (SERA) ecosystem and tested using robot EMotive headY System (EMYS) on a novel scenario called Quick Numbers regarding likeability, intelligence, animacy, and proximity. There is no statistical significance on the obtained results, however, from the video footage, we noticed that the participants manifested more positive reactions and emotions in the rapport condition, therefore, a sample higher than 20 might reveal the expected statistical results. Finally, researchers may integrate the rapport model on any SERA agent with low effort.

Keywords
Human-Robot Interaction (HRI); Rapport; Rapport; Framework; EMotive headY System (EMYS)

1. INTRODUCTION
Robots are increasingly becoming part of our society and their presence has been proven to impact our lives. But do any of us remember a remarkable interaction with a robot to the same degree we are able to recall one with a person? What makes one conversation memorable?

In order to answer these questions, researchers have been exploring agents capable of building rapport, i.e., designing agents that consider the following aspects during interactions: positivity, mutual attention, and coordination [45]. In reality, most of the today’s agents do not consider formally this concept, and yet they have been impacting people’s lives on several scenarios such as education [7], child care [6] and medical assistance [20, 26].

Researchers have been mostly considering single aspects of rapport such as gaze [1, 2, 5, 30, 46, 53] and backchannel [12, 17, 31, 34, 35, 49] (listener behaviour), which is not enough, as rapport is managed more efficiently when considering its three components. More importantly, we only found virtual agent capable of managing these three components [15, 18]. This work tackles the lack of robotic agents capable of building rapport using its three components. For this purpose, we designed and implemented a rapport model (Sections 4 and 5, respectively) on robot EMYS using the SERA ecosystem [38]. The designed rapport model enables concurrent execution of rapport behaviours using a prioritisation mechanism where idle actions (e.g., postural mimicry) have lower priorities than behaviours triggered momentarily (e.g., surprise animations). These strategies may be either rule-based or Machine Learning (ML)-based but they all cooperate to achieve the interactional goals of the scenario.

To analyse the performance of the developed rapport strategies, we conducted user studies to study its impact on likeability, intelligence and liveness using Godspeed questionnaires [4, 25], and proximity [3] using the Quick Numbers scenario detailed in Section 8.

2. RAPPORT
The feeling of flow and connection during interactions is formally known as rapport [52]. Spencer Oatey further suggests that rapport management can be divided into three main objectives: build, maintain or destroy [45]. These goals are only be accomplished by manipulating the three components of rapport suggested by Tickle-Degnen and Rosenthal [48] (Figure 1):

- **Positivity**: the feeling of approval and friendliness (e.g. humour and self-disclosure);
- **Mutual attention**: the feeling that the other’s attention is focused on the individual (e.g. mutual gaze);
- **Coordination**: the feeling of predictability and being in-sync (e.g. postural mimicry).
Above all, it is essential that the three components of rapport co-exist during interactions \[10, 14, 53, 54\]. Naturally, the relative weights of these components, as well as the strategies may change as the relationship evolves beyond that of strangers \[10, 53, 54\].

### 3. RELATED WORK

Regarding computation models of rapport, Zhao, Papangelis, and Cassell, propose a theoretical model to manage long-term rapport \[32, 54\] that follows the planning concept depicted in Figure 1. The model considers adapting the agent’s behavioural responses according to the current nature of the relationship that is continuously in evolution. In addition, researchers have noted the importance of being able to continuously adapt to the interaction, give incremental feedback \[23, 24, 37, 50, 55\], and even recover from mistakes \[19\]. This requires incremental planning and execution of behavioural chunks that can be potentially interrupted, modified, and even replaced \[24, 37, 50, 55\].

Lastly, researchers have been developing models for particular strategies of rapport such as gaze \[1, 2, 5, 30, 33, 44, 46, 53\], and backchannel \[12, 17, 29, 31, 34, 35, 49\]. These systems are either rule-based \[1, 16, 19, 27, 31, 38, 49, 52\], ML-based \[8, 9, 11, 12, 22, 28, 30, 47\], or even, more recently, hybrid \[18, 41, 42\]. In particular, the developed framework, that supports the rapport model described in Section 4, is built using the Socially Expressive Robotics Architecture (SERA) ecosystem as it provides the required tools to develop robotic agents.

#### 3.1 Socially Expressive Robotics Architecture

Socially Expressive Robotics Architecture (SERA) follows asynchronous messaging architecture similar to Robot Operating System (ROS) \[36\] where clients communicate with one another through Thalamus network messages. Beside supporting robotic agents, SERA provides non-technical researchers the ability to customise the agent’s behaviour using mark-up text (Table 1).

The most important elements of SERA are:

- **Thalamus**: receives and delivers the published messages to the right subscribers;
- **Skene**: translates high-level intentions into actions;
- **Nutty Tracks**: animation engine;
- **Speech Server**: Text-To-Speech (TTS) engine.

Effector clients, in order to change agent’s behaviour, have to contact Skene that processes and redirects the request to the most suitable clients: Nutty Tracks for animations and Speech Server for utterances. However, despite supporting interruption of actions, SERA is not without issues as:

- If a speech request arrives while there is one in execution, the Speech Server executes them sequentially which can cause unintended repetition of utterances;
- There is a noticeable (less than a second) delay between the requests and the effects, as requests pass firstly through Skene, Thalamus and finally, either Nutty Tracks or Speech Server.

#### 4. RAPPORT COMPUTATIONAL MODEL

The model attempts to build rapport following the goal tree illustrated in Figure 1. It is sufficiently generic and customisable to tailor the agents to different embodiments and/or HRI scenarios. Furthermore, the model supports interruption and replacement of actions according to the dyadic state of the interaction, as it is vital to adaptive agents \[24, 37, 50, 55\]. For example, stop speaking at any time to give the turn to others to speak, possibly apologising for taking the time for doing it \[18\].

Following Figure 2, the rapport model has the following components:

- **Dyadic State**: contains information regarding the interactional partner and the environment;
- **Perceptions**: perceptual information;
- **Rapport Effectors**: components that generate signs of rapport by proposing actions to the Rapport Controller given the dyadic state of the interaction;
- **Rapport Controller**: manages the received actions proposals.

![Figure 1: Example of a goal tree to build rapport. The nodes are goals and the leafs are actions.](image)

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>greet</td>
<td>Hi</td>
</tr>
<tr>
<td>game</td>
<td>score</td>
<td>Yeah! &lt;Animate(surprise2)&gt;</td>
</tr>
<tr>
<td>game</td>
<td>results</td>
<td>Managed &lt;Points&gt;! &lt;gaze(person)&gt;</td>
</tr>
<tr>
<td>end</td>
<td>ending</td>
<td>I am glad to have met you! &lt;animate(happy)&gt;</td>
</tr>
</tbody>
</table>

Table 1: Set of utterances compatible with Socially Expressive Robotics Architecture (SERA). Actions are delimited by < and >, and substitution variables by |.
Figure 2: Schematic representation of the rapport model.

4.1 Rapport Controller

The Rapport Controller manages the proposed actions sent by the different rapport strategies. Each action proposal is a quintuple $A = \langle G, P, E, I, T \rangle$ containing:

- **Group** ($G$): part of agent’s body that the action is attempting to manipulate;
- **Priority** ($P$ where $P \in \mathbb{N}_0$): relative importance of the action proposal in relation to others;
- **Execution description** ($E$): how the agent will execute the action;
- **Interruption description** ($I$): how the action should be interrupted by the agent;
- **Timeout** ($T$ where $T \in \mathbb{N}$): maximum duration.

Concerning the management of action proposals, the controller periodically captures a snapshot of the agent’s ongoing actions and pending action proposals received by the different Effector. Whenever a snapshot is captured or, when receiving a new action proposal, the controller analyses which actions should be interrupted (and replaced).

As long as two action proposals, $A$ and $S$, have different groups ($A_G \neq S_G$), both will execute using $A_E$ and $S_E$. However, as illustrated in Figure 3, in the case of a conflict ($A_G = S_G$), the action with the highest priority ($A_1$) will interrupt and replace the current one ($A_2$). Moreover, the action can be time limited, otherwise, if the action takes longer than expected ($T$) it is interrupted, despite having higher priority. As rule of thumb, idle actions should have a lower priority than actions triggered by discreet states.

4.2 Stimulate Positivity

In pursuance of enhancing the first component of rapport, the Positivity rapport Effector takes into account the dyadic state of the interaction to trigger vocalisations to motivate, praise, and even humour the interactional partner (Table 2). In addition, the agent should share a personal information to the person (self-disclosure) that plays an important role in closing relationships between two strangers [21].

<table>
<thead>
<tr>
<th>Interactional State</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>After Introduction</td>
<td>I learnt the casting move today!</td>
</tr>
<tr>
<td>Player loses the queen</td>
<td>Your king is still well guarded!</td>
</tr>
<tr>
<td>Agent loses a bishop</td>
<td>There goes a bishop!</td>
</tr>
</tbody>
</table>

Table 2: Example utterances depending on the current state of the interaction.

4.3 Stimulate Mutual Attention

Following Figure 1 and 2, the model enhances mutual attention using mutual gaze and backchannel strategies. The Mutual-Gaze rapport Effector bases on the work developed by Andrist et. al. [1]. In short, it swaps between establishing eye contact and looking the game during predetermined periods of time according to the following conditions: current phase of the scenario (in-task or between-tasks) and the user’s personality (introverted or extroverted). The researcher may set the priority of the action proposals, as well as the gaze lengths, which default to the values specified on the work this Effector is based on [1].

The Backchannel Effector bases on the work developed by Niewiadomski et. al., on the GRETA system [31] that analyses variations of the pitch to produce listener behaviour. For simplicity, this Effector only considers up-down head nods as listener signals. The researcher may set the priority of the generated action proposals, in addition to the trigger probability, the amplitude of the gesture, the gesture velocity, and the number of head nods.

4.4 Stimulate Coordination

To enhance coordination, the model uses behavioural mimicry and backchannel strategies.

The Facial Expression Mimicry Effector mimics facial expressions (e.g., happiness, surprise, and sadness), triggering animations according to the perceived emotion intensity $I \in [0, 100]$. In addition to the priority of the generated action proposals, the model allows changing the following parameters for each type of emotion: trigger probability, minimum intensity, and priority.

Head Gesture Mimicry Effector mirrors head gestures such as up-down nods and left-right shakes [2,10,39,52]. In addition to the priority of the generated action proposals, for each gesture, researchers may specify the amplitude, the velocity, and the number of repetitions.

Lastly, the agent should adhere to social norms by, for example, introducing itself when meeting people and saying “thank you” when a person does a task for the agent.

5. EMYS: THE RAPPORT AGENT

The following section describe the framework that implements the rapport model detailed using EMotive headY System (EMYS) robot (Figure 4) as the chosen embodiment to test the rapport agent.
The framework follows a plugin-based architecture that, in order to reduce the impact of network delays (Section 3.1), the internal communication is done using method calls, limiting the number of network messages. Following Figure 5, the framework contains the following elements:

- **Effector Plugins**: proposes actions and enables/disables plugins;
- **Perceiver Plugins**: perceives the external world and informs the interested plugins;
- **Utility Plugins**: general purpose plugins that can be used to, for example, store interactional data;
- **Rapport Controller**: manages plugins’ lifecycle, link plugins, and has the same responsibilities as the rapport model’s Rapport Controller (Section 4.1);
- **Thalamus Connection**: bridges the system with SERA, sending and receiving Thalamus messages.

The *Cooperation* connection depicted in Figure 5 establishes the connection between plugins, for example, *Effectors* use *Perceivers* to read perceptual information and may use *Utility* plugins to consult the interaction history. In addition, the *Action* connection is the decisive message that will execute the action proposals defined by the *Effectors* and executed by the *Rapport Controller*.

### 5.1 Plugins’ Lifecycle Management

At the startup, the *Rapport Controller* loads the available plugins from a user-selected folder. They are all enabled by default unless specified otherwise through the configuration file that can be accessed using the controller’s Graphical User Interface (GUI) (Figure 19 in Appendix A). During this process, following Figure 6, each plugin follows a two-step initialisation:

1. **Initialisation**: initialise internal variables;
2. **Retrieve Dependencies**: retrieve plugins that it depends on (e.g., *Effectors* typically requires *Perceivers*).

After initialisation, if the *Rapport Controller* is running, *Effectors* can attempt to modify the agent’s behaviour concurrently according to the dyadic state of the interaction, until they are disposed of either manually (Figure 19 in Appendix A), either automatically by another *Effectors*. In any case, the plugins are automatically disabled in case of an error without escalating to an application crash.

### 5.2 Managing Actions

The *Rapport Controller* manages action proposals as described in Section 4. However, in addition to the elements of action proposal quintuple $A = \langle G, P, E, I, T \rangle$, the controller stores the following information:

- **Status**: pending, executing, executed or interrupted;
- **Starting time**: when the action has started executing;
- **Thalamus identifier**: to monitor when actions have finished by monitoring the Thalamus messages sent by Nutty Tracks and Speech Server.

The status field is required to manage the state of each action proposal. For example, following Figure 7, an action proposal is only interrupted (using $I$) if its previous state was *executing*. In the absence of the *Thalamus* identifier, we could not flag the actions as executed, making them transition automatically to the interrupted state.

To conclude, the execution and the interruption descriptions of the action proposal (E and I, respectively) are self-contained functions specified by the researcher, therefore they can contain additional logic. However, they must at least send the required *Thalamus* messages to either *Nutty Tracks* or *Speech Server* to produce animations or utterances, respectively.
### 6. PLUGINS

There are three types of plugins: **Effectors**, **Perceivers**, and **Utility**. Transparently to the developer, each plugin may specify its GUI, and more, importantly, its configuration file (e.g., Listing 1) that allows researchers to tailor the agent’s behaviour to the interactional goals of the scenario without having to recompile the code.

**Listing 1: Excerpt of the XML configuration file used by Mimic Facial Expressions Effectors.**

Researchers may additionally define a high-level **Effectors** that is capable of deactivating and enabling other plugins in runtime, giving the agent the ability to map the rapport strategies to discreet states of the scenario.

We also noticed that some **Effectors** frequently interrupt themselves repeatedly. To solve this issue, we added, transparently to the developer, a mechanism to track internally the proposed actions by specifying the minimum delay between each action proposals. The researcher can choose one of the following levels:

- **Unrestricted**: the **Effectors** must explicitly manage its proposed actions;
- **One Action Globally**: the **Effectors** cannot interrupt itself unless with a proposal with higher priority;
- **One Action Per Group**: same as **One Action Globally** but with granular to the **Group**.

#### 6.1 Agent Actions Manager

The Agent Actions Manager is a **Utility** plugin with the following goals:

- Monitors agent’s actions to notify the controller;
- Provide convenient wrappers for common action proposals, describing both **E** and **I**;
- Provide non-technical researchers with the tools to change the agent’s behaviour without worrying about implementation details.

The first objective is achieved by monitoring the messages that SERA sends to the **Thalamus Network**. The second objective aims to reduce the amount of code required to specify common action proposals. To accomplish the last goal, similar to **Skene**, this plugin provides a GUI (Figure 20 in Appendix A) to change the agent’s behavioural rules (utterances) using mark-up text (Table 3). Taking advantage of the rapport model, following Expression 1, these utterances may contain interruptible or time-limited actions with different priorities.

\[
\text{< action(arg1, arg2, ..., argn, [priority], [delay], [timeout]) >}
\]

#### 7. EFFECTORS IMPLEMENTATION

The following section describes the technologies that support the rapport **Effectors**, their parametrisation and the encountered issues.

The Positivity rapport **Effectors** (that satisfies the “Stimulate positivity” goal) and the “Adhere to Social Norms” goal are implemented by triggering utterances given the perceptual information of the scenario.

**7.1 Supporting Technologies**

Revisiting the rapport model described in Section 4, the rapport **Effectors** needs to monitor the dyadic state and perceive the user’s emotion, speech disfluencies and head gestures. To this end, following Figure 8 there are four main components to support the rapport strategies:

- **Social Signal Interpretation (SSI)**: framework to recognise social signals in real time \(^1\) [51];
- **SHORE**: recognise facial features from a video feed [40];
- **openSMILE**: extracts prosody features [13];
- **GRETA\textsuperscript{PP}**: adapted version of GRETA [31] to generate listener behaviour and redirect perceptions;
- **GRETA Perceiver Plugin**: proxy between GRETA\textsuperscript{PP} and the Rapport Controller.

![Figure 8: Schematic representation of the components that supports the rapport model.](http://hcm-lab.de/projects/ssi/)

\(^1\)http://hcm-lab.de/projects/ssi/
GRETA_PPP is a variation of the GRETA system [31] that contains solely its Listener Intention Planner rule-based component that behaves identically to the rapport model’s Backchannel Effector. The perceptions and the generated behaviour are sent to the GRETA_PPP Perceiver Plugin using User Datagram Protocol (UDP) sockets so that it notifies the interested Effectors. The system has a slight delay (< 1 second), as the refresh rate had to be halved to 5Hz so that the system could handle the resource heavy SSI.

The following Effectors’ parameters were parametrised empirically following several pilots on 3 different people. The generated action proposals have the priority of 1, with the exception of Backchannel Effector that has priority of 2 as it is not considered an idle behaviour. Finally, the mimicry behaviour is only done once every 3500ms.

7.2 Facial Expression Mimicry
The Facial Expression Mimicry rapport Effector mimics the user’s emotion given SHORE’s emotion with highest intensity. With the probability of 50%, it proposes actions, as long as happiness, sadness, and anger intensities detected are 0.65, 0.65, and 0.5, respectively. Nonetheless, the implementation of this plugin is not without issues as SHORE was detecting happy emotions in absence of faces. We solved that issue by applying a smoothing signal at the cost of increasing the delay from less than 1 second, to at most 3 seconds. In the end, we opted to remove the smoothing filter and compensate with disabling it when the user was not present.

7.3 Head Gestures Mimicry
The Head Nod Mimicry Effector proposes up-down nods and left-right head shakes given Kinect’s perceived head gesture. As the sensors seldom detected head gestures during the pilot tests, both gestures’ probabilities are set to 1. In addition, it produces light head nods that are slightly randomised in each action proposal to avoid being repetitive.

7.4 Mutual-Gaze
The Mutual-Gaze Effector swaps the gaze target depending on the current state of the interaction assuming that the user is extroverted (lowest standard deviation), as we are unable to know the personality beforehand. The gaze durations are the default values.

7.5 Backchannel
The Backchannel Effector produces listener behaviour according to the GRETA system [31], proposing up-down head nods actions. However, we were unable to reduce the impact of noise (agent’s robotic movement and voice) on the SSI sensors, leading to excessive false positives. Despite the attempts of counteracting the issue through noise-suppressing directional microphones and adjusting the sensors’ parameters, the issue persisted, therefore the Backchannel Effector was not used during the studies.

8. STUDIES
The developed rapport system was tested using the Quick Numbers scenario (Figure 9) regarding likeability, intelligence and liveness using the Godspeed series [4, 25], and Proximity [3].

![Figure 9: An example of a participation in the Quick Numbers study (side view).](image1)

8.1 Quick Numbers
In the Quick Numbers scenario players are tasked with gaining as many resources as possible within the given time. At the beginning, each player starts with a fixed amount of resources that can be invested before each round. Depending on the amount invested and the player’s performance, the returning investment will either be greater or lesser than the investment. The task is to tap the appearing numbers in sequential order starting with number one until the end of the round (Figure 10). With each successful tap, the score increases, and with each incorrect number, the score decreases. In the end, the resources earned are the product between the amount invested and the round’s score.

In this study, EMYS accompanies the subject throughout the scenario, not as an opponent, but as another player that is also playing the game. The scenario stages go as follow:

- **Introduction**: brief explanation of the rules, followed by the start the scenario;
- **Training Stage**: the participant plays an informal match alone to get accustomed to the game mechanics;
- **Gaming Session**: both players play a single round, at the same time;
- **Results Discussion**: the agent comments the each player scores;
- **Investment**: the participant is informed that he has to invest on the agent;
- **Ending**: EMYS informs the value of the investment return and thanks to the subject for his participation.

![Figure 10: Illustration of the Quick Numbers game developed in Unity.](image2)
8.2 Quick Numbers Plugins

In addition to the plugins that implements the rapport model, we developed the following:

- **Scenario Perceiver**: monitors the scenario through Thalamus, notifying the interested Effectors;
- **Utterances Manager**: proposes utterances according to the dyadic state of the interaction;
- **Rapport Strategies Manager**: enables/disables the rapport Effectors throughout the scenario;

The Utterances Manager Effector implements the Positivity rapport Effector and satisfies the “Adhere to Social Norms" goal of the rapport model by proposing utterances tailored to Quick Numbers (Table 3). As the participant speaks Portuguese, different utterances are used depending on the subject’s gender.

The Rapport Strategies Manager Effector disables postural mimicry and mutual gaze rapport strategies during the Gaming Session and Investment stages, when EMYS participates in Quick Numbers as a player and not as a spectator. In particular, it disables Facial Expression Mimicry Effector in the investment phase because the participant’s face is not visible on the video feed.

8.3 Results

A group of 40 participants from different universities campus participated in this study. The participants were equally selected and randomly distributed between the control condition (C) and the rapport condition (R), making two independent groups. Condition C has a mean age of 23.5±1, equal distribution of genders, and over 55% of the participants already interacted with EMYS. Condition R has a mean age of 25.65±3.945, 60% male and only 25% of the sample played with EMYS in the past.

The study follows a between-subjects design, using significance level of 5% for every statistical analysis.

**Are the rapport strategies effective in making agents more likeable?**

From the questionnaires, we compared the likability statistics between condition C (Figure 11) and R (Figure 12). As the distributions do not follow a normal distribution on conditions C and R are not normal (sigC = 0.002 and sigR = 0.003), therefore we compare both conditions with the Mann–Whitney U statistical test. As the histograms’ shapes (Figures 13 and 14) are distinct (sig = 0.512), we can only compare the mean scores: 4.25±0.716 and 4.3±1.031, on conditions C and R, respectively.

**Answer**: Despite the greater man average on the rapport condition, there is no statistical proof that rapport strategies are effective on increasing likeability. However, given that the mode (most frequent value) changed from 5 to 6, from condition C to condition R, we can postulate that, if we increase the sample size, we might obtain a clear confirmation that the rapport strategies affect the agent’s likeability.

**Does the developed system improve agent’s liveness?**

Similar to likeability, the animacy distribution on both conditions does not follow a normal distribution (sigC = 0.002 and sigR = 0.003), therefore we compare them using the Mann–Whitney U statistical test. As the histograms’ shapes (Figures 13 and 14) are distinct (sig = 0.512), we can only compare the mean scores: 4.25±0.716 and 4.3±1.031, on conditions C and R, respectively.

**Answer**: There is no definite proof that the current system improves the agent’s liveness.

**Do rapport strategies affect the perceived intelligence?**

Similar to likeability and animacy, the perceived intelligence distributions on conditions C and R are not normal (sigC = 0 and sigR = 0), therefore we compare both conditions with the Mann–Whitney U statistical test. The histogram’s shapes (Figures 15 and 16) are dissimilar (sig = 1.0), therefore we only compare the mean scores: 5.1±0.553 and 5.0±0.918, on conditions C and R, respectively.

**Answer**: There is no statistical proof that the rapport strategies have an effect on the agent’s perceived intelligence.

**Are the rapport strategies effective in establishing closer relationships between humans and agents?**

In order to answer the last hypothesis, we compare the proximities answers between conditions C (Figure 17) and R.
(Figure 18) using a 7-item scale question. In both conditions, proximity follows a normal distribution \( \text{sig}_C = 0.203 \) and \( \text{sig}_R = 0.304 \), therefore we use the independent \( t \)-test statistic test which yielded the statistical significance value of 0.694 \( (> 0.05) \), consequently we reject the null hypothesis.

![Figure 17: Proximity histogram in the control condition.](image1)

![Figure 18: Proximity histogram in the rapport condition.](image2)

**Answer:** Despite the small increase of 0.2 on the mean score from condition C to R, the impact of the rapport strategies on proximity are inconclusive.

### 9. DISCUSSION

The developed robotic rapport agent did not reveal statistically significant results between the control and the rapport condition, given the subjective measurements collected from the questionnaires. We believe that the lack of statistical significant result was due to the low sample size \( n = 20 \) in each condition) and due to the fact that the participants were more focused on the task and not on the agent. Notwithstanding, participants showed more positive reactions in the rapport condition. On one occasion, the participant remarked the agent’s capability to synchronise his happy animation with its laugh. One another two independent occasions, both male and female participants flattered the agent’s ability to distinguish their gender. Furthermore, it is also possible to retrieve behavioural metrics such as smile frequency and eye contact duration, by annotating the recorded video (1235 minutes) to look for additional metrics that may confirm that the rapport model improves current agents.

### 10. CONCLUSION AND FUTURE WORK

In this paper, we presented the design and implementation of a rapport model that enables robotic and virtual agents to show natural signs of rapport on any HRI scenario according to the dyadic state of the interaction. The resulting framework was built on top of the SERA ecosystem and tested using robot EMYS on the Quick Numbers scenario.

Overall, the system reveals to be a success as we are currently able to reuse the rapport model on any SERA agents with low effort. In addition, we made the first steps on building richer agents by integrating the SSI and SHORE on our system which increases the agent’s perceptual capabilities. Furthermore, we encourage researchers to design rapport agents, as it is noticeable from the experiments that it possible to bring forth more positive emotions by using dedicated rapport strategies.

In the future, we propose using a sample size greater than 20, use participants with no previous experience with the agent and improve the scenario to provide more quality rapport opportunities. In addition, we should look into both behavioural metrics such as eye contact and smile frequency. The rapport model can be improved by considering mechanisms to assess rapport success so that the agent might adapt its strategies to the interaction, and even recover from mistakes [19]. For example, use SHORE to monitor the emotion state of the user and attempt use humour to cheer him as soon as sadness becomes the most average emotion. The model should also explore resumable actions, that is, explore agents capable of resuming their course of action after being interrupted by external stimuli. Finally, the backchannel Effector should be revisited as it is only lacking an improved noise suppressing mechanism.

### 11. REFERENCES


[41] M. Schröder. The SEMAINE API: Towards a


**APPENDIX**

**A. FRAMEWORK SCREENSHOTS**

![Figure 19: GUI representation of the system.](image1)

![Figure 20: GUI representation of the Agent Action Manager Utility plugin.](image2)