Implemeting a trust mechanism for collaborative AI agents

Alexandra-Ioana Neagu(5233194)
[1]TU Delft
A.I.Neagu@student.tudelft.nl

1 Introduction

This report details the trust mechanism implemented by Group 62 for interdependence relationships in the MATRX search and rescue environment. Specifically, RescueBot's trust mechanism aims to efficiently maximize task completion. To accomplish this, we implemented a trust mechanism based on the perceived willingness and competence of a human agent whose characteristics are unknown. This mechanism is integrated into the decision-making protocol of the RescueBot to optimize collaboration between the agents.

The trust mechanism is evaluated in three collaboration scenarios consisting of a human agent with differing *willingness* and *competence* characteristics. The evaluation is further compared to three baseline implementations, and further recommendations are provided based on these results.

2 Trust Mechanism

The trust mechanism is based on the idea that an artificial agent working with a human whose characteristics and conditions it is not aware of, will infer a measure of trust in regard to this human based on the perceived *willingness* and *competence* they show during collaboration or when performing individual actions. As such, we have categorized certain interactions as affecting the perceived competence (e.g. if a human collects a mildly injured victim alone, this means that they must not be weak, so the competence of this human is increased), or the perceived willingness (e.g. if a human does not respond within a certain amount of time to a call for help from RescueBot, their willingness will be reduced). Together, these two measures are used to determine the decision-making behavior of RescueBot, with the goal of optimizing task completion.

The measures are updated after every interaction, such as a received message or collaborative action, in addition to a maintained confidence level in the current trust belief. Based on its beliefs, RescueBot will adapt its behavior depending on whether an action can be completed **independently**, or if it is tied to the human agent as either a **soft** or **hard** dependency.

The design of our trust mechanism is discussed in section 2.1. The subsequent sections 2.2 and 2.3 detail how the trust belief is updated, and how this influences the decision-making behavior of RescueBot, with respect to perceived willingness and competence.

2.1 Code Architecture

Given the provided decision-making protocol in *OfficialAgent.py*, the agent's trust logic was refactored to *trust.py*. This contains a singular *Trust* class, which models the trust with respect to one agent, maintaining their perceived willingness and competence, as well as the RescueBot's confidence in those values. While multi-agent collaboration is outside the scope of this paper, this would allow for a convenient extension of the decision protocol to evaluate trust with several agents.

After the refactoring of the *trust.py* class, the trust functionality left in the *OfficialAgent.py* class relates to how this trust mechanism is taken into account by the agent when making decisions and interacting with the human. The conditions of the human are not known to RescueBot, so we heavily modified its response process to integrate this trust system for more efficient collaboration between the agents.

Variables

The willingness and competence values maintained in the *Trust* class are restricted to the range of $[-1, 1]$ (as per the assignment description), where negative values indicate unwillingness and incompetence. The variables are initialized to those provided to the agent in *allTrustBeliefs.csv*, or 0 if it is the first time collaborating with this agent.

In addition, the RescueBot's confidence in those values is maintained as a multiplier variable in the range $[0, 1]$. This is initialized to 0.1 at the start of every round. The characteristic variables are multiplied by the confidence variable. So, when confidence is low, e.g. at the start of rounds, the RescueBot's decisions will be more lenient towards the collaborating agent.

Methods

The *Trust* class provides two methods to evaluate the trust of the agent within the decision-making protocol of *OfficialAgent.py*: `willing` and `competent`. After experimenting with different thresholds, it was decided that an agent is willing or competent if its variable for the perceived characteristic exceeds the `LOW` threshold of -0.2. A marginally negative threshold was chosen as this allows some room for fluctuations in the factors. This is necessary because it is not possible to directly observe the truthfulness of other agents' interactions, and thus it is likely there will always be some error in the perceived characteristics.

Additionally, there are methods for updating the perceived characteristics of the collaborating agent. These methods are called in *OfficialAgent.py* after every interaction, such as a received message or performed action, and they increase the agent's confidence in their current trust belief. The update from messages method that checks each received message from the human also calls the aforementioned methods to update competence, willingness, and agent's confidence.

It is worth mentioning that confidence is updated in a multiplicative manner, i.e. by 30% of the current confidence. This models trust more accurately: the agent gradually builds confidence up over time; and, if its confidence is high and the collaborating performs unexpectedly, a 30% drop in this value is significant.

Another important method that this class contains is the waiting method, which maintains how long the agent has been waiting for the human to perform an expected action, and in the case of a timeout, lowers the willingness value. A timeout value of 300 game ticks was adopted based on the dimensions of the grid, the proportion of water tiles, and the fact that a human agent cannot reply while performing an action.

Lastly, it is worth noting that our methods provide console log messages when the RescueBot's behavior is modified, or when trust values are updated. This helps introduce transparency into its trust system and how it updates throughout the course of a round. With this, we strived to create an artificial agent that follows the principles of explainable AI (XAI), furthering the trust in this human-AI system.

2.2 Perceived Willingness

This sub-section discusses how the perceived willingness of the collaborating human is updated, and how this affects the decision-making behavior of RescueBot.

- When the human sends a message about searching a room, asking for help to remove, finding a victim, or rescuing a victim, this indicates the agent is proactive and willing to play the game. Thus, **increasing willingness** by a small amount.

  However, if the human sends several messages searching different areas without finding any victims or obstacles, they may be lying, and thus the agent **decreases confidence** in their current trust belief.

- If a victim is found as the human told, the agent **increases willingness**. Otherwise, they have lied or rescued the victim without informing the agent, thus, **decreasing willingness**.

- On finding a tree, the robot waits for 300 game ticks on a response. On receiving a response, the agent **increases willingness** by a small amount. If the human does not reply within this time, it defaults to removing the tree, and **decreases willingness**. In addition, if the human is not willing, the agent skips this wait. The Continue response is only accepted if the human is willing. The Remove response **increases willingness**.

  This is because removing a tree is an independent action: the human cannot support the RescueBot. Thus, it can be generally assumed that removing a tree will always be beneficial toward round completion.

- On finding a big rock, the robot waits for 300 game ticks on a response. On receiving a Remove Together response, it waits for an additional 300 game ticks for the human to arrive and remove the object. If the idle waiting time of the agent runs out, it defaults to the Continue response and **decreases willingness**, as there is nothing it can do. If the action is successfully carried out, the agent **increases willingness**.

  This is because removing a rock is a hard interdependence. It is a collaborative action requiring both parties to be present.

- On finding a critically injured victim, also a hard interdependence, an identical approach is taken on the Rescue Together and Continue responses.

- On finding a mildly injured victim, the robot waits for 300 game ticks for a response. On receiving a Rescue Together or Continue message, the RescueBot will default to Rescue Alone if willingness is low. Otherwise, it waits for an additional 300 game ticks for the human to show up. Again, if time runs out, the RescueBot **decreases willingness**, and defaults to performing the action alone. Otherwise, if the action is successfully carried out, the agent **increases willingness**.

  This is because rescuing a mildly injured victim is a soft interdependence relationship: human assistance improves efficiency. However, as the human may be lying about showing up, or simply take a while to respond, the RescueBot can infer this unwillingness and complete the action independently.

- On finding a small rock, also a soft interdependence, an identical approach is taken on the Continue, Remove Alone, and Remove Together responses.

2.3 Perceived Competence

This sub-section discusses how the perceived competence of the collaborating human is updated, and how this affects the decision-making behavior of RescueBot.

- On receiving a message stating that the human is collecting a victim, this indicates that they are not weak. Thus, the agent **increases competence**.

- On rescuing a mildly injured victim, if the human asks for help from RescueBot to collect the victim together, the agent **decreases competence**, as it may indicate that the human is weak.

- On receiving a duplicate message that the human is searching the same room again, this indicates that they are unable to keep track of which rooms they have already searched. Thus, the agent **decreases competence**.

- If the human is not competent, the RescueBot waits for 400 game ticks instead of 300 for the human to arrive or to respond to a message. This is because a fully competent human (strong human) can carry all mildly injured victims together, can detect obstacles from far away, and take actions accordingly, which makes them faster and more responsive. Thus, it can be generalized that an incompetent human might need more time to respond/arrive at a location even though they are willing.

The provided decision-making logic in *OfficialAgent.py* checks for the characteristics of the human and modifies its behavior based on that. However, we were asked to not check for these characteristics, and not modify the state of the agent in those places. This made it challenging to find areas to incorporate competence without violating the assignment requirements.

3 Evaluation

This section documents the strong and weak points of the agent and analyzes its collaborative strength individually and in comparison to the three baselines (*NEVER_TRUST*, *ALWAYS_TRUST*, and *RANDOM_TRUST*). Lastly, further improvements are provided based on the results of this analysis.

3.1 Considered Types of Human

For evaluating our implemented trust mechanism, we have considered the following 3 types of humans: Alice, Ben, and Charlie.

Alice

She is a *strong* human. This means she can carry all mildly injured victims at the same time and detect obstacles from further away (10 grid cells, as opposed to 1). She is also truthful: she always completes the actions she tells RescueBot she is doing; replies to the RescueBot's requests in time; and does not lie when requesting help for carrying a victim or removing an object.

So, Alice represents the willing and competent human agent archetype. When impersonating her, we ran the environment with the 'strong' condition and were simulating a fully truthful way of playing, where we never lied to the agent about the actions/observations she was performing.

Ben

He is a *lying* human. This means that his information sharing does not always match actions and/or observations. As an example, he could be lying about having searched an area, about finding a victim, or an obstacle.

So, Ben represents the unwilling, yet competent human agent archetype. When impersonating him, we ran the environment with the 'normal' condition, but we were simulating an untruthful environment, purposefully lying to RescueBot about the actions we were performing 80% of the time (4 out of 5 actions).

Charlie

She is a *weak* and *lazy* human. Her weak character means that she is unable to carry mildly injured victims or remove the small brown stone alone, requiring assistance from RescueBot. In addition, due to her lazy character, she may abandon her previously communicated action before completion and start another one. However, she is truthful about which actions she is performing. In addition, she may take a long time to respond to messages from RescueBot.

So, Charlie represents the unwilling and incompetent human agent archetype. When impersonating her, we ran the environment with the 'weak' condition and were simulating a way of playing where we would abandon 50% of the tasks we communicated to RescueBot (1 out of 2 tasks), and also delay our responses to its messages about 1 in 2 times, with different delay times.

3.2 Performance over Several Rounds

First of all, to have a full understanding of how well the trust mechanism works, three consecutive rounds are played for each type of human. The completeness progress of each round is plotted and compared for the three humans. The *Tables 1-3* include the values related to time to finish the task, number of actions, and at what ticks the maximum completeness is achieved for each round.

Alice	Run 1	Run 2	Run 3
Total ticks	3763	3932	3052
Human actions	1305	1387	837
Agent actions	2380	2459	1906
Max completeness at tick	3763 (1.0)	3932 (1.0)	3052 (1.0)

Table 1: Round information for Alice

Ben	Run 1	Run 2	Run 3
Total ticks	6000	6000	6000
Human actions	351	790	1190
Agent actions	2757	2794	3459
Max completeness at tick	3365 (0.25)	3359 (0.375)	4912(0.5)

Table 2: Round information for Ben

Charlie	Run 1	Run 2	Run 3
Total ticks	6000	6000	6000
Human actions	952	631	679
Agent actions	1912	2502	2033
Max completeness at tick	4716 (0.375)	5631 (0.5)	4712(0.5)

Table 3: Round information for all Charlie

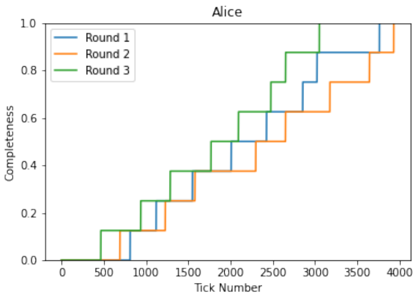

Figure 1: Completeness of the three rounds for Alice

In the case of Alice, as she is strong and willing, she gains trust quickly, so the prior trust beliefs about her do not majorly impact her performance in the next round. Because of this, even from the first round, she manages to complete the round in less than 10 mins. Subsequently, her performance improves, and she achieves 100% completeness quicker in each round.

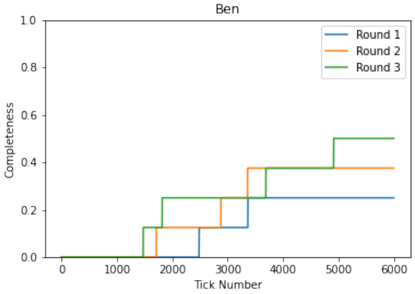

Figure 2: Completeness of the three rounds for Ben

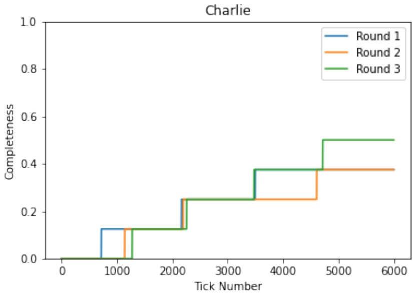

Figure 3: Completeness of the three rounds for Charlie

In the case of Ben and Charlie, due to their untrustworthy nature, rounds become very lengthy and it takes more than ten minutes to achieve completion, all rounds with them ended at 6000 ticks due to the game timeout. For Ben especially, the game gets stuck as the victims behind large rocks cannot be rescued as RescueBot never waits for Ben to help remove it because of his very low willingness value. However, it can be still observed that overall, they achieve higher completion in the same amount of time in Round 3, as the RescueBot learns to cooperate with them better when the game starts with more accurate trust beliefs. Whilst playing, it is observed that since Ben is lying the majority of the time, he cannot finish the game in 10 minutes, yet his untrustworthiness makes RescueBot collect 2-3 mildly injured victims and 0-1 critically injured victims, as it learns to stop listening to Ben and perform tasks independently, landing him around 0.25-0.5 completeness. For Charlie, she is lazy and weak, and her competence value decreases over rounds. As Charlie claims to have searched areas and does not afterward, and asks help to carry even the mildly injured victims, Charlie has a low competence value after three rounds. Her willingness, however, fluctuates between 0 and 0.650 as she is willing 50% of the time.

At the end of the 3 rounds, the final trust values of the human agents are:

Human Agent	Competence	Willingness
Alice - Run 1	0.300	1.0
Alice - Run 2	0.799	1.0
Alice - Run 3	0.900	1.0
Ben - Run 1	0.0	-0.399
Ben - Run 2	0.0	-0.599
Ben - Run 3	0.1	-0.799
Charlie - Run 1	-0.300	0.449
Charlie - Run 2	-0.4	0.650
Charlie - Run 3	-1.0	0.0

Table 4: Competence and willingness values for all rounds

It can be observed that even though the RescueBot does not directly know the characteristics of the three humans, it is able to understand them according to direct experiences, as we can see it reflected in the trust values. Alice, a truthful human, reaches the maximum willingness of 1.0 from the start as she always reacts positively to RescueBot's cooperation attempts, and her competence improves with each round as the agent grasps her ”strong” condition, also converging at the end to 1.0. Ben's competence stays consistent around 0.0, as the agent learns he is neither ”weak” nor ”strong”, but his willingness goes down with each round due to his lies. Lastly, Charlie's competence belief devolves into negative values because of her ”weak” condition, eventually converging to -1.0. Her willingness level is fluctuating in between 0.0-0.65, which is sub-optimal, and indicates that the willingness belief calculations are not perfect for a lazy human. Nevertheless, Charlie's willingness value is greater than Ben's, since Charlie only shows unwillingness in 50% of the time, as opposed to 80% for Ben.

Overall, the implemented trust mechanism models conditions related to competence well, converging within a few rounds to a value that accurately represents the ”strong”, ”normal”, or ”weak” condition of the human, which are 1.0,0.0, and -1.0 respectively. However, for willingness, while the mechanism models willing human and lying human accurately, it does not model willingness with the same amount of efficiency and accuracy for lazy human. It takes more rounds for willingness values to converge to an optimum, and they present more fluctuations. This is due to the inconsistent nature of the ”truthfulness” condition of a human. A ”strong” or ”weak” human is always strong or weak, but a lying human does not lie consistently, or a lazy human responds late in different amounts. Also, it is straightforward to check whether a person is lying or not, by, for instance, checking if the victim is where it is supposed to be, however it can be relatively more difficult to detect laziness.

3.3 Comparison against Baselines

Furthermore, the trust mechanism is tested for each type of human against the 3 evaluation baselines, and analyzed using line charts. For the comparison between evaluation baselines and the original mechanism, the third rounds from the previous sections are used.

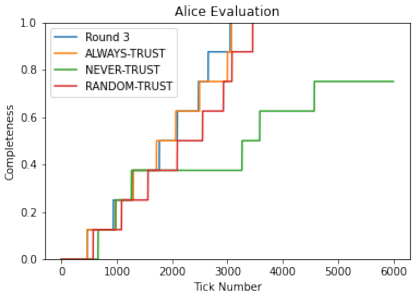

Figure 4: Alice against evaluation baselines

When impersonating Alice, the *ALWAYS_TRUST* baseline achieves completeness almost the same way as our trust mechanism. This is because Alice is a fully willing and competent human, thus trusting her from the beginning is very efficient. This shows that the mechanism is working as intended. *RANDOM_TRUST* baseline achieves completeness similarly, only a bit slower. On the other hand, the *NEVER_TRUST* baseline cannot achieve full completeness because, in our trust implementation, the agent skips critically injured victims and big rocks if the human is not willing, which may result in the agent being stuck on a cycle if there are no small rocks, mildly injured victims, or trees left. Normally, as trust is built gradually, and willingness can be improved by searching rooms, finding victims, etc. the agent does not get stuck in a cycle. In this case, as trust values are always -1, even though the human is willing and competent, the RescueBot does not listen to the human at all and cannot complete the actions that require collaboration. This points to a weakness in the implemented trust mechanism: when there are no mildly injured victims, small rocks, and trees left, if the willingness of the human is very low, the agent might not perform any action after that point, thus 100% completeness might not be achieved. As a further improvement, the agent can wait next to a critically injured victim or a big rock that needs to be carried together and communicate with the human to come over.

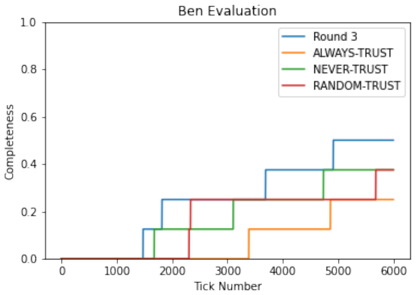

Figure 5: Ben against evaluation baselines

When impersonating Ben, the *ALWAYS_TRUST* baseline resulted in the worst completeness level, as expected, since Ben is a lying human, and fully trusting him will always lead to inefficient cooperation. The *NEVER_TRUST* provided the most similar result to our trust mechanism, again as expected, due to Ben being an untruthful human.

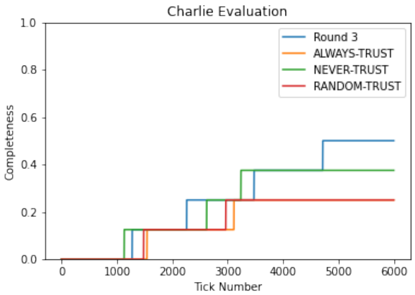

Figure 6: Charlie against evaluation baselines

When impersonating Charlie, our trust mechanism reaches to a completeness level greater than all 3 baselines. Since Charlie is both unwilling and incompetent, our mechanism behaves radically better than *ALWAYS_TRUST*. It also behaves better than *NEVER_TRUST*, because Charlie still shows willingness in about 50% of cases, and our system adapts to this.

Overall, our implementation achieves higher completion than the 3 baselines, as the artificial agent's behavior constantly adapts based on how it learns to trust the human. An important feature that we introduced that contributes to its efficiency is the *waiting* method. Now, the agent does not remain idle forever waiting for a response from a possibly unwilling human and starts doing tasks independently in such situations. Furthermore, it takes into account such idle waiting times and reduces willingness based on them.

3.4 Further Improvements

Considering the shortcomings we treated above, here we present possible improvements to our trust mechanism. The way willingness is modeled could stand to be improved the most, so a first potential future improvement could be, when confidence in trust is low (the human has been proven to be unreliable), the agent can follow the human to see whether their messages correspond to their actions, and update willingness based on that.

We can also improve how the tree removal action affects willingness further by treating the following case: if a human asks for help in removing a tree but upon arrival, the agent sees there is no tree, willingness is decreased, as the human is perceived as having lied about it.

Last but not least, we can still improve the competence modeling more. A possible extension for the mildly injured victims and the small brown stones could be to check for distance. If we assume that ”strong” and ”normal” humans only ask for help if the robot is nearby, we can reduce competence only if the human is far away.