The Effect of Theory of Mind in Detecting Deception

Rui Miguel Barata Castanho Soares

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. João Miguel de Sousa de Assis Dias
Prof. Ana Maria Severino de Almeida e Paiva

Examination Committee
Chairperson: Prof. Daniel Jorge Viegas Goncalves
Supervisor: Prof. João Miguel de Sousa de Assis Dias
Member of the Committee: Prof. Rui Filipe Fernandes Prada

May 2016
First I would like to thank my advisors, Prof. João Dias and Prof. Ana Paiva, for guiding me and giving me the tools to make it through this big challenge. I also want to thank all my colleagues in the research project PArCEIRO, for all their help and support. I want to thank GAIPS for giving me a great environment full of nice people to work with.

Finally I thank my family and closest friends for always being there for me when I needed.
Abstract

Social simulation is a research area that has been gradually increasing, and systems that can simulate human behaviour are more and more necessary. One of the main capacities for social behaviour is Theory of Mind, the ability to represent not only ours, but others’ internal knowledge, such as their beliefs. Theory of Mind influences aspects like deception and its detection. Therefore, this thesis proposes an agent capable of detecting deception through the use of Theory of Mind. The architecture was based on a research on the state of the art, together with some preliminary tests, with humans playing the game. The agent was implemented as a robot, and used to play a deception game called Coup with humans. Finally, this thesis presents user studies with two conditions, one agent with Theory of Mind and one without it. The results revealed our Theory of Mind agent is capable of detecting more lies from human players than the one without Theory of Mind.

Keywords: Lie Detection, Theory of Mind, Social Agent, Coup, Human-Robot Interaction
Resumo

A simulação social é uma área que tem vindo a crescer gradualmente, e sistemas capazes de simular comportamento humano são cada vez mais necessários. Uma das principais capacidades para o comportamento social é a Teoria da Mente, a abilidade de representar não só conhecimento interno mas também o dos outros. A Teoria da Mente influencia aspectos como a decepção e a detecção da mesma. Esta tese propõem um agente inteligente capaz de detectar decepção a partir do uso de Teoria da Mente. A arquitectura é baseada numa pesquisa sobre o estado da arte, assim como em testes preliminares com humanos a jogar o jogo. O agente foi implementado como um robot e usado para jogar um jogo de decepção chamado Coup com humanos. Para terminar, esta tese apresenta estudos com utilizadores com duas condições, um agente com Teoria da Mente e um sem. Os resultados mostram que o nosso agente com Teoria da Mente é capaz de detectar mais mentiras de jogadores humanos do que o agente sem Teoria da Mente.

Palavras-Chave: Detecção de Mentira, Teoria da Mente, Agente Social, Coup, Interacção Humano-Robot
# Contents

List of Tables xi  
List of Figures xiii  

1 Introduction 1  

2 Background 3  
2.1 Deception 3  
2.2 Lie Detection 3  
2.3 Theory of Mind 4  
2.4 Mindreading Model 4  
2.5 Coup, The Game 6  

3 Related Work 9  
3.1 Theory of Mind Approaches 9  
3.2 Theory of Mind in Deception 10  
3.3 Theory of Mind in Robotics 12  
3.4 Theory of Mind in Human-Robot Teams 13  
3.5 PsychSim 14  
3.6 Study on Reasoning about Others 15  
3.7 Summary 16  

4 Preliminary User Tests 19  
4.1 The Session 19  
4.2 Results 20  

5 Theory of Mind Agent 23  
5.1 The Architecture 23  
5.2 Theory of Mind Component 24  
5.2.1 Theory of Mind Level 2 26  
5.3 Deliberative Component 27  
5.3.1 Accommodating Theory of Mind 28  

6 EMYS, the Coup Player 31  
6.1 Overall System Architecture 31  
6.2 EMYS Robot and Speech Component 32  
6.3 Digital Tabletop and Coup Unity Game 34
List of Tables

2.1 COUP: summary of characters, actions, action effects and counteractions . . . . . . . . . . 7
4.1 Number of times participants lied about having a character . . . . . . . . . . . . . . . . . 20
4.2 Times players challenged each other and times they were right . . . . . . . . . . . . . . . . 20
7.1 Lies in the Theory of Mind condition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
7.2 Lies in the No Theory of Mind condition . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
A.1 Utterance’s categories and subcategories explained . . . . . . . . . . . . . . . . . . . . . 54
List of Figures

2.1 Baron-Cohen’s Theory of Mind’s model ................................................. 5
3.1 Agent model for Theory-theory (on the left) and Simulation-theory (on the right) . . . 10
3.2 Model with one-level Theory of Mind proposed for the agent proposed by Dias et al. . . 11
3.3 Agent sets tree structure by Felli et al. .................................................. 16
5.1 Our Agent Architecture ........................................................................... 23
6.1 Overall System Architecture ................................................................. 31
6.2 FATIMA core architecture ................................................................. 32
6.3 EMYS ..................................................................................................... 33
6.4 Agent architecture with the utterance manager module and player information ....... 33
6.5 Coup game’s original interface ............................................................. 35
6.6 Original interface’s “Yes” (Sim) and “No” (Nao) buttons for challenging phase ......... 35
6.7 Coup game’s improved interface .......................................................... 37
6.8 Cards showing with “Hide button” (“Esconder”) ........................................ 37
6.9 Pressed help button showing the “cheat sheet”. ........................................ 38
6.10 Improved interface buttons for challenging phase (“I doubt it” and “I believe”) ....... 38
6.11 Improved interface buttons for counteracting phase (“I block” and “I do not block”) . 38
7.1 Godspeed’s dimension of perceived intelligence from the first and third sessions ....... 41
7.2 3rd session means from question 5 “I think that EMYS perceived when I lied” ........ 42
7.3 3rd session means from question 3 “I think that EMYS was capable of deceiving me throughout the game” .............................................................. 43
7.4 3rd session means from question 6 “I think that EMYS played in a competitive way” . 43
7.5 3rd session means from question 13 “EMYS was easily deceived by me” ................. 44
7.6 3rd session means from question 10 “I liked playing the game with EMYS” ............. 44
7.7 1st session means from question 15 “I was bored with the course of the game” ......... 45
Chapter 1

Introduction

Imagine a world where you could not tell, or at least suspect, when someone was lying to you, where you would blindly believe everything you were told. It is hard to imagine, right? Well, some people live in that world and computers do too.

Even though many intelligent agents have assumed a sincerity condition[1] in which they naively believe there is no deception, this is not true for every situation, especially if we want to portray human behaviour and their interaction. It is a known truth, people lie. They lie out of love, out of fear, in order to get something or keep something (e.g. a relationship), etc. there are a lot of reasons for lying[2]. So assuming the need for agents to be capable of lying, they should also be capable to detect such lies from others.

We can see deception as the act of one transmitting information that he believes not to be true, or has no idea to be true or not, for the purpose of misleading. So in order to detect these lies, we need to know, or at least speculate about, what are the beliefs of the person doing the deception. Let's see a simple example:

John invites his friend Bob over for dinner and his mother does codfish. Even though John believes Bob does not like the dish because he never seen him eat it at school and while Bob is eating, John can see from his reaction he is not enjoying it, when John's mother asks Bob if he liked it, he replies “yes”. John knows it is a lie that Bob tells because he believes John's mother will find him impolite if he says “no”.

What John just did is possible due to a thing called Theory of Mind, that is the ability to recognize and attribute mental states not only to ourselves but to others, and understand that beliefs we have may be different from others.

According to studies on Theory of Mind, humans start developing this sort of “mind reading” at a young age, but this may not be true for all humans. A research published by Simon Baron-Cohen, Alan M. Leslie and Uta Frith[3], suggests that children with autism lack a Theory of Mind and therefore have difficulties understanding another person's beliefs. This results in an impairment of their social behaviour and interaction.

Intelligent agent’s behaviour is still far behind human standards, and one of the reasons is most of them also lack a Theory of Mind. Which also leaves them with an impairment when put to test on social simulations. This impairment is noticeable in their inability to lie or detect lies.

Even though we can find some works on Theory of Mind applied to areas like robotics or even deception in agents, we cannot find relevant studies on its use to detect the latter. We consider this to be as important because an agent unable to detect lies “living” in a world where deception exists is vulnerable.
Problem

We want to create agents with the ability to detect lies told in a social simulation. To do this on a general context would take too much time and resources. In order to test this on a feasible scale, we will use the game COUP.

COUP is a social game based on deception, where players lie in order to make other players lose influence (face-down cards) while maintaining theirs. In this game, when players believe they are being lied to, they can challenge the plays from other players. This makes COUP a perfect context for deception and its detection studies.

We will create an agent that can play the game, focusing on the task of detecting deception. So, taking this into consideration, this thesis tries to answer the following research question: How to create intelligent agents that are able to detect human players’ deception in a game of COUP?

Hypothesis

In Coup we can doubt almost all of the players’ actions but doing so is risky and we may lose cards. So we need to guess what other players are thinking. In order for an agent to be able to do this, it needs some kind of representation of other’s minds. This can be achieved through the use of the previously introduced concept, Theory of Mind.

We believe that an agent equipped with Theory of Mind shall detect other players’ lies more often while playing the game COUP against a human than one without Theory of Mind.

Organization

In Chapter 2 we have the Background where theoretical concepts will be introduced for understanding of the work to be done. Together with a full explanation of the game COUP.

In Chapter 3 we find the Related Work, where work related to the topic will be presented and compared.

Chapter 4 presents the Preliminary Tests done with humans. Starting with an explanation of the test, proceeded by the conclusions taken from its results.

In Chapter 5 we present our agent’s architecture and its components, most importantly we explain how we endowed it with Theory of Mind.

Chapter 6 is where we explain the overall architecture and how all the components and their integration.

Our Evaluation is presented in Chapter 7. We explain the tests we did in order to confirm our hypothesis and present our results followed by a discussion of these.

In the end there is a conclusion and thoughts about future work.
Chapter 2

Background

In this section we provide explanations for theoretical topics relevant for our work, and also insight to the game COUP.

2.1 Deception

Deceiving can be seen as one person changing what the other thinks into something that whoever is deceiving believes to be false, for whatever purposes. For example, researchers sometimes purposely mislead or misinform participants about the true nature of the experiment. Like in the experiment conducted by Stanley Milgram in 1963[4], the researchers told participants that they would be part of a scientific study on memory and learning. In reality the study looked at the participants’ willingness to obey commands, even when those involved inflicting pain upon another person. So we could say the researchers knowingly introduced false beliefs in the participants minds so they would not suspect what they were being tested for. Another example of this mind changing ability is bluff, the pretense that your position is stronger than it really is. In bluff games inducing of false beliefs is quite common. A player may do some action, knowing it will lead to other players thinking a certain thing, like one player raising the bet in poker may lead to other players thinking that player has a good hand. So people use this type of reasoning to fool others into thinking something that is not true, in the given example of poker, players raise the bet even though their hands are not that good.

2.2 Lie Detection

Detecting lies would be an easy job if everyone was like Pinocchio[5], but since that is not true and noses just do not get any bigger from lying, other methods need to be used in order to detect such lies. Such methods can go from the analysis of a person’s voice[6] to the more tortuous injection of truth serum[7]. The most known method for lie detection is the Polygraph, which measures physiological indices of a person being interrogated that are then compared to differentiate the lies and truths told. Although this is somewhat effective[8], like some other lie detection methods, we cannot use it in every occasion, especially to detect everyday life lies, since we need the technology to use it. But there are other methods humans use on their quodian to detect lies that do not require such technologies, like body reading (shaking your head negatively while saying yes, swallowing, facial micro-expressions etc.[9]) or mind reading. Although reading the term mind reading takes us to think about telepathy, that is not the case, this is a less fictitious but nevertheless fantastic skill most humans are capable of thanks to Theory of Mind.
2.3 Theory of Mind

The term Theory of Mind was first used on a study by David Premack and Guy Woodruff[10] and refers to the ability to infer mental states, such as intention, knowledge, belief, thinking and so forth, not only to ourselves but also to others, and understand that other's mental states may differ from our own. It is called a theory because we have no way of observing the mind, we can say thoughts are there, but we cannot really see them.

There is a big philosophical debate concerning two approaches to Theory of Mind: theory-theory and simulation-theory. In theory-theory[11], Theory of Mind is considered to be implicit and it is acquired by the automatic maturation of an innate module. The theory comprises beliefs, desires and plans, and basis on which these interact. It allows the understanding, explaining and prediction of self and other’s behaviour. Mental states of others are unobservable and therefore only knowable by intuition and insight. Simulation-theory[12] on the other hand sees Theory of Mind as the ability to mimic the mental state of another person, by casting ourselves into the other person’s mind and simulating his actions with our own reasoning.

Overall, Theory of Mind creates an empathy that unites us as a society, we are always appraising others’ thoughts and feelings, not only to understand and predict their behaviour, but also to adapt our own in response to theirs. We use Theory of Mind every day without even noticing, for example: “I think she likes strawberries”, “I suspect he is bluffing”, “I think he believes I don’t like him”. The last example is different from the other ones because it has two levels of Theory of Mind. A one-level Theory of Mind can represent what another person is thinking, a two-levels Theory of Mind will represent what another person thinks a third person is thinking, and so on and so forth for the other levels. The higher the level, the more complex Theory of Mind becomes.

Studies show we’re not born with a fully developed Theory of Mind but it is instead developed during our childhood[13]. To test human children for the presence of Theory of Mind, psychologists have been using a false-belief method for quite some time[3]. This method consists on presenting the child with the scenario:

Sally takes a marble and puts it in her basket. She then exits the room and goes for a walk. While she is away, Anne takes the marble out of Sally’s basket and puts it in a box. Sally then comes back and wants to play with her marble.

The child is then asked the question: "Where will Sally look for her marble?". The answers to this question depend on the age of the child. Children around three years old will answer “In the box”, because they know the ball is there but they do not yet have the notion that other’s knowledge may differ from their own. Children that are a little bit older, around 4 or 5 years old, will answer “in the basket”, even though they know the ball is actually in the box, they are capable of distinguish what others know, from their own knowledge. We can say that children around this age have a Theory of Mind, even if only capable of one-level assessments.

Even though this development is true for most people, there are some disorders, such as autism, that cause Theory of Mind impairments that result in poor social behaviour and interaction skills. Baron-Cohen calls this lack of capability to attribute mental states, to the self and other, Mindblindness[14].

2.4 Mindreading Model

Simon Baron-Cohen has come up with a model that explains how humans use Theory of Mind to read into others’ minds[14]. The model, in figure 2.1, is based on 4 mechanisms that he says to be roughly
representative of four properties of the world: volition, perception, shared attention, and epistemic states. Although there are other models, like Leslie’s[15], we chose to present this one due to its modules connections being better specified.

The first mechanism is the Intentionality Detector (ID) and its job is to translate other’s actions into primitive mental states, such as goals and desires. ID is based on senses (vision, touch and audition) and it interprets almost anything with self-propelled motion, or making a non-random sound, in such terms as: Wants(X, Y) and HasGoal(X, Goal), with X and Y being entities.

The Eye-Direction Detector (EDD) works through vision and has 3 main functions: (1) detecting the presence of eyes or eye-like stimuli; (2) computing whether eyes are directed towards it or something else; (3) inferring that if the eyes are directed towards something, then that entity sees that something. This last function allows for perceptual states to be attributed by the observer, such as “Bob sees me” or “dog sees cat”.

Both this mechanisms representations, Wants(X,Y), HasGoal(X,Goal) and Sees(X,Y), can be referred as dyadic, in the sense that they only define the intentional relationship between two entities.

So that triadic relations, between three entities, are possible, the third mechanism is necessary. This mechanism is called Shared Attention Mechanism (SAM), and its work is making such triadic relationships with information from the first two perceptual mechanisms. SAM fuses dyadic perceptions from another’s perceptual states and dyadic perceptions from self into triadic representations. Thus it is possible to have states in the form Sees(X,Sees(Y,W)) like: “I see that Mario sees the door”. Evidences show that when SAM links EDD with ID, eye-direction can indicate desires and goals[16].

The last mechanism is called Theory of Mind Mechanism (ToMM) and is based on Leslie’s theory. It is a system that, as the name may suggest, employs the Theory of Mind. ToMM’s work is divided in two main functions. Its first function is transforming the information received from the SAM into epistemic mental states of the form Attitude(Agent, Proposition) like Believes(Bob, “it’s sunny”). These representations were baptized by Leslie as M-Representations. We should note that in this representations, the proposition may be false and the state still true, e.g. it may be raining outside but since Bob has his curtains shut, he still believes it is sunny. The second function of ToMM is tying this mental states together into a usable theory that can be helpful in reasoning, predicting and influencing actions from others.
2.5 Coup, The Game

Coup is a board game played with cards and coins which runs on bluff and deduction. It was published by Indie Boards and Cards in 2012. The game can be played by 2 till 6 players. In the beginning of a game each player starts with two coins, taken from the treasury, and two cards, taken from the court deck. The game’s objective is to make others lose their cards while keeping ours. When someone loses a card they must turn it up and it remains like this until the end of the game, with the player not being able to use it.

The game is played in turns, each turn the player can do one of the actions from table 2.1. There are seven actions: (1) Income, where the player takes one coin from the bank; (2) Foreign-aid where instead of one, the player takes two coins, but there is a downside that we will explain ahead; (3) Coup, in which the player pays seven coins and chooses a player to lose a card (when losing a card, it is the player who is losing it that chooses the card to lose from the two, of course if there is only one faced-down card there is no choice to be made), if a player starts his turn with 10 or more coins, he is required to do Coup; Then we have actions that are associated to some character, this characters are: the Duke, that is able to do (4) Tax, in which the players gets three coins from the bank; the Assassin, that allow us to do pay three coins and do (5) Assassinate to another player, which has to lose a card; the Ambassador, that allows the player to (6) Exchange, this is, to trade cards, the way the trade is done is by taking two cards from the deck and mix them with the facedown cards, then choosing the cards to keep and returning two cards to the deck and shuffling it; the Captain gives the player the possibility to (7) Steal from other players, taking two coins from a player it chooses; finally we have the Contessa which has no unique action, but if we look back to table 2.1 you can see it has a thing called counteraction, in this case, Contessa blocks assassinate, if a player does assassinate to another, that second player can block with the Contessa, keeping its cards. There are two more counteractions, the block steal, which can be done by the Captain and Ambassador and keeps the player who did Steal from taking the two coins, and the Block Foreign-Aid associated to the duke, this is the downside of the foreign aid mentioned previously, if someone blocks it you get no coins that turn. The first two counteractions can only be done by the player on the receiving end of the actions, while the third one, Block Foreign-Aid, can be done by anyone when a player uses foreign aid.

But Coup is a bluff game, and as in any other bluff game we can lie about what we have. Seen that the cards are faced-down, only we know our characters, so we can use any action from any character. Of course there is a risk, because other players can doubt that we have that character. For example, player A does Assassinate on player B, but for some reason player B does not believe A has the Assassin so he challenges the action, so player A needs to show the Assassin. If the player does not have the card upon being challenged, he needs to lose a card of his choice and the action goes through. But challenging a player is not risk free, in the given example, if player A has indeed the Assassin, player B loses the challenge and needs to lose one card, plus the action has no effect. Players can also lie about counteractions, so even if player B did not have the Contessa, he could still block the Assassinate and of course, player A could challenge that. There are only three cards of each character, this is important to find out if a player is lying since if there are two faced-up Dukes and I have the third in my hand when a player does Tax I know for sure he is lying so it is a safe challenge.
<table>
<thead>
<tr>
<th>Character</th>
<th>Action</th>
<th>Effect</th>
<th>Counteraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Income</td>
<td>Take 1 coin</td>
<td>x</td>
</tr>
<tr>
<td>-</td>
<td>Foreign Aid</td>
<td>Take 2 coins</td>
<td>x</td>
</tr>
<tr>
<td>-</td>
<td>Coup</td>
<td>Pay 7 coins</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose player to lose 1 card</td>
<td></td>
</tr>
<tr>
<td>Duke</td>
<td>Tax</td>
<td>Take 3 coins</td>
<td>Blocks Foreign Aid</td>
</tr>
<tr>
<td>Assassin</td>
<td>Assassinate</td>
<td>Pay 3 coins</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose player to lose 1 card</td>
<td></td>
</tr>
<tr>
<td>Ambassador</td>
<td>Exchange</td>
<td>Take 2 cards from deck, return 2 cards to deck</td>
<td>Blocks Stealing</td>
</tr>
<tr>
<td>Captain</td>
<td>Steal</td>
<td>Take 2 coins from another player</td>
<td>Blocks Stealing</td>
</tr>
<tr>
<td>Contessa</td>
<td>x</td>
<td>x</td>
<td>Blocks Assassination</td>
</tr>
</tbody>
</table>

Table 2.1: COUP: summary of characters, actions, action effects and counteractions
Chapter 3

Related Work

This section presents systems and studies that are somehow related to what is intended with this project system.

3.1 Theory of Mind Approaches

Herbes et al. paper’s goal is modeling agents with a Theory of Mind[17]. They believe that systems with intelligent virtual agents can help people training for complex and dynamic tasks, giving as example crisis management, firefighting or military missions. The use of those virtual agents instead of humans makes the training more flexible and may reduce personnel costs. But for this to be possible the agents need to be a good replacement for those humans, hence needing to show believable behaviour and interact with the users in a human way. So instead of only taking into account their own goals and beliefs, agents should also take into account assumed knowledge and intentions of others. They need a Theory of Mind.

To model and implement the agents with Theory of Mind they propose two approaches inspired in the philosophical theories explained in Chapter[2] Theory-theory (TT) and Simulation-theory (ST). Since both theories can be represented in terms of beliefs, goals and intentions, the BDI paradigm was used for modeling both theory’s agents.

The TT agent model is represented on Figure 3.1 on the left. It is composed of: (1) goals, by which the agent is directed; (2) plan library, which contains the plans for achieving those goals, given certain preconditions; (3) intentions, the commitment to go through the plan; (4) beliefs, of its own and others; (5) a reasoner, that upon receiving a sense, does all the reasoning, considering all the other components, in order to output an action. Besides having beliefs about others, the agent has also other’s assumed goals and reasoning rules that it uses to reason over such beliefs.

The ST agent model has as a main difference the fact that since the agent makes use of all its reasoning power when reasoning about others, it does not need all the other’s reasoning steps. The model is represented on Figure 3.1 on the right and has a reasoner, just like the TT agent, its own and other agent’s mental states. The mental states are composed of beliefs, goals, plans and intentions (the same as in TT) and represented as different modules, this way the agent can use the same reasoner to reason with different mental states independently.

A scenario was created where two agents needed to communicate with a human in order to extinguish a fire. Then some changes were made to the scenario to see how the agents would react when the human or the other agent made mistakes. Three agents were tested in this scenario (and its variations), one without Theory of Mind, one with TT and one with ST. The results show agents acted as expected
Figure 3.1: Agent model for Theory-theory (on the left) and Simulation-theory (on the right)

and that agents with a Theory of Mind, contrary to the ones without, were able to provide explanations of the consequences of their actions, support actions and mistakes. No differences were spotted between the results of the TT agents and the ST agents. Nevertheless implementation-wise, ST allows for code reuse, for structural errors of agents and can be applied to numerical representation of agent models. All things that TT did not allow, making ST a better option for more complex agent models.

3.2 Theory of Mind in Deception

Dias et al. work revolves around Theory of Mind (ToM) on agent’s deception, more particularly, in how a higher level of Theory of Mind on agents, influences their believability compared to lower or no levels of Theory of Mind\[18\][19]. They believed that an agent that can only represent what another is thinking (1 level of ToM) is less believable deception-wise, than one that not only can represent what the other is thinking, but also what the other thinks others are thinking (2 levels of ToM).

They used their research to create deceptive agents for an interactive game with virtual characters. Their objective with this was to prove that “The higher the level of reasoning about the others a virtual agent is capable of using, the better and more believable it can perform deceptive tasks”.

Their approach was based on the Theory of Mind model by Baron Cohen, and makes use of a BDI model approach of Simulation-Theory, alike Meyer et al. approach (Chapter 2.4). They call this agent a “Mindreading Agent” and its proposed model is presented in Figure 3.2. The agents perceive events and update their Knowledge Base (KB), which stores the agent’s beliefs and knowledge of the world, and the Theory of Mind module (ToMM). This module is divided in the three components from Baron-Cohen's approach (explained in detail in the previous chapter): the EDD, SAM and ToMM. Leaving the ID out due to the high complexity of the intention recognition area.

In order to cope with more levels of ToM, the Model of Others (constituent of the ToMM) was built as a recursive structure. Each Model of Other had its own ToMM with Models of Others. This is a complex model, where the more ToM levels, the less efficient each update cycle is. For this reason they decided to focus on only two levels for this paper.

The EDD and SAM determine how a given perception will be used to update the models of others in the ToMM. The EDD determines what entities, objects, and perceptions are perceived by other agents by allowing the making of domain specific rules that restrict the perceptions. This makes it possible for actions to have a global effect, where they are assumed to be perceived by everyone, or a local effect, where only specific agents perceive them. When EDD receives a perception, with the use of this rules, it will produce two lists, the perceptionVisibilities and the agentVisibilities. The first list indicates which agent(s) perceive the received perception, while the second list states what agents are seeing
which agents. The SAM will then use this two lists’ information to determine what Models of Others get updated. It does this by checking for each model, if it is in perceptionVisibilities, then it checks for each of the predecessors of that model its presence in the agentsVisibilities, if both check out, then the model can perceive the perception, otherwise the algorithm stops following the remaining sub-tree and continues the recursiveness.

To deliberate upon what action it will take, the agent has a Deliberation module that does all the planning considering both KB and ToMM’s information. It does this through the use of two mechanisms, a forward oriented one based on inference rules and a backward oriented one used to create plans of actions to achieve the agent’s goals.

An inference rule has a set of conditions that need to be true in order for the rule to be applied, and a set of effects that will modify the KB when the rule is applied. The deliberation module tests the preconditions of the rules, if a rule is applied, it then proceeds to change the KB accordingly to the rule’s effects. If the KB is modified this way the Deliberation will repeat this process until no more changes are observed.

Goals have also preconditions, when this preconditions are verified, the goal becomes active and the second mechanism of the Deliberation component starts building a plan of actions to achieve a list of success conditions for that goal.

The initial step for the Deliberator to account for the ToM information was allowing the specification of preconditions P to be tested with a Model of Other instead of the agent’s KB. This is done by giving the preposition P a list of agents: $A_{g1}$,...,$A_{gn}$,P. For example: A:B: Suspects(A) will be true if Suspects(A) is true in the Model of Other B that is kept by the Model of Other A. To test the preposition against the agent’s own KB it just needs to be written without any agent before it, just P.

The second step was to make it possible to model rules and goals to change the mental state of others. For this they used the same mechanism used to establish local and global effects. An effect is described as $A_{g1}$,...,$A_{gn}$,P, where $A_{gi}$ is an agent’s name or the symbol “*”, which means that “all Models of others at that particular level will be selected”. Only the list of Models of Others on the $A_{g1}$,...,$A_{gn}$ list will have added to their KB the preposition P.

If an inference rule has an effect with an agent’s list, it will not update its own KB, instead it will go over the Models of Others’ structure and update it according to the list. Models of Other’s will also be checked for updates by the inference mechanism at every update cycle.

In order to study their approach, Dias et al. tested their model with non-player characters in a social game based on deception called MIXER, where a group of villagers has to discover who among them is the werewolf, this last one must keep his true identity “hidden”. The villagers who played the role of victims were all implemented with one level of ToM. The werewolf on the other hand had two versions,
one with one level and the other with two levels of ToM. Their results showed that the werewolf with two levels of Theory of Mind produced better results in the game than one with just one level. They then conducted online questionnaires to find out how users perceived their two agents. The users’ answers showed that the two levels of ToM agent lied better, being less predictable and more difficult to the victims to catch, it was also perceived as more intelligent.

### 3.3 Theory of Mind in Robotics

In this paper Scassellati presents two theories, Leslie[15] and Baron-Cohen [14], on the development of Theory of Mind in children, discussing the possible use of both in building robots with a Theory of Mind[20]. Theory of Mind is referred as “the ability to correctly attribute beliefs, goals and percepts to other people”, or “ability to mentalize”.

Scassellati believes that a robot equipped with a Theory of Mind will have social interaction capabilities, that would not be possible otherwise, and be capable of learning through observation the same way children do. The robot will be able to express emotions, desires, goals, and recognize these in others, reacting in a more accurately way to them, learning also how to anticipate reactions from others and modify its own behaviour appropriately.

A problem with the implementation of both Theory of Mind models is the amount of components and the required level of coordination between them. The two models describe perceptual and motor skills that are necessary for the more complex Theory of Mind capacities. The most notable differences between them are the way in which they divide the perceptual tasks. Since Leslie’s model makes a clean division of the perceptual world into animate and inanimate objects, whereas Baron-Cohen provides more detail on limiting the perceptual input each module requires.

The initial system focused on two abilities: distinguishing between animate and inanimate motion and identifying gaze direction. This robotic system they built was called Cog and consisted on a pair of six degree-of-freedom arms, a three degrees-of-freedom torso, and a seven degree-of-freedom head and neck. It had a visual system made out of four color CCD cameras, two microphones made up the auditory system, the vestibular system consisted of a three axis inertial package, and a collection of kinesthetic sensing from encoders, potentiometers and strain gauges.

Cog has pre-attentive visual routines that resemble those of human infants, with three basic feature detectors: color saliency analysis, motion detection and skin color detection. These together with the habituation effects, are combined with weights from motivations, drives and emotions, and filtered through the visual attentional system for more complex post-attentive processing like face recognition.

The face recognition system is designed to find areas that are likely to contain a face. It focus on locations with skin color and/or movement, which are then passed to an algorithm called ratio templates which works well with changes both in illumination and slight rotational ones. A sub-image containing the eyes can be extracted from the peripheral image of locations that are classified as faces by the algorithm. Baron-Cohen’s EDD first function is matched by these functions, which can also begin to approach the second and third function.

A second system discriminates between animate and inanimate visual stimuli, operating at two developmental stages, and based on the existence of self-generated motion. On a first stage only spatio-temporal features are used for tracking. More complex object features, like color, shape and texture, are left for the second stage. This systems provide the assumptions that the ID of Baron-Cohen requires.

In order for having gaze following, a system capable of detecting eye contact, also needs the ability to extract the angle of gaze, to extrapolate the angle of gaze to a distal object, and motor routines for alternating between the object and the individual. The ability of gaze following in infants is developed
during their first year of living for distal objects inside their field of view, and some months later starts gaze following objects outside of the field of view. The gaze following of the robot can be achieved with a geometric strategy and refined with feedback from the individual gazing the object. The representational stage however, will require the robot to maintain information on salient objects that are outside of the field of view, like appearance, location, size, and salient properties.

Scassellati finishes his paper with suggestions for adapting the gaze following solution to deictic gestures like pointing, and also extending the animate-inanimate distinctions which have many flaws.

### 3.4 Theory of Mind in Human-Robot Teams

The work developed by Hiatt et al. had as a goal building human-robot systems to be used in open, real-world environments. Their paper describes the design of a robot for such system, capable of dealing with the human behaviour of its teammates through the use of Theory of Mind[21].

The Theory of Mind was used to account for human variability during team operation, for this, their approach simulates human thinking by executing cognitive models of those same humans. Cognitive models are an approximation to cognitive processes for the purposes of comprehension and prediction, and are based in experimental data of human behaviour[22]. So a robot tries to identify what different beliefs, intentions or desires led the human teammate to deviate from the team goals by putting itself in the human's shoes.

The architecture chosen to make this project was ACT-R[1], one of the most popular architectures for cognitive modeling. For developing the approach they used the robotic simulation environment Stage[2] as the robot's "world".

The actual approach accounts for two possible sources of variability when trying to explain the human's behaviour. The first one is at model level, which is not deterministic, so there are different paths and each one is assigned a probability through the probabilistic simulation analysis process. The second is associated with different hypothetical cognitive models. A few of the differences to the models that may originate distinct behaviour are differences in knowledge and beliefs about the world, in subgoals or in parameterizations, being that their work ignores the last ones.

The probabilistic simulation analysis analyzes hypothetical models of human teammates. Through a slight modification of ACT-R's structure of execution, it considers all possible execution paths that could occur, and also each path's associated probability.

There are two types of branch points during execution: when there is more than one chunk matching a memory retrieval request, and when more than one production rule can fire at a certain moment. When this branch points are reached, the probability of each branch being followed is calculated. After, all the branches will be followed in order, starting with the first, until all possible branches have been explored and all possible paths, with their associated probabilities, returned.

In the beginning the robot has no way of generating hypothetical models. But when it observes behaviour he cannot explain, it asks the human to explain such behaviour. He remembers the human's answers and next time the human does something unexpected it will test if this newly learned reason explains the human's behaviour. This variances are then given different priorities based on recency and frequency. Therefore, a variance that happens more often and has been used recently will be tested before one that does not happen so often and is older. If variances have an extremely low frequency and have not been used recently, they end up being forgotten.

So basically the robot executes the paths of its model and if no path leads to the observed action

---

1 [http://act-r.psy.cmu.edu/about/](http://act-r.psy.cmu.edu/about/)
2 [http://playerstage.sourceforge.net/stage/stage.html](http://playerstage.sourceforge.net/stage/stage.html)
of the human, it proceeds to create different hypothetical behaviors communicating its understanding of the situation to the human. If no model leads to that action the robot simply asks the human what he is doing, saving the answer for later.

In order to test the approach, two scenarios were made in which the human teammates acted in an unexpected way. For each scenario, three different approaches were recorded: the Theory of Mind approach and two other, the first one with a simple correction approach, where the robot would always warn the human every time he did some unexpected move, and the second with a blindly follow approach, where as the name suggests, the robot would simply follow the human and never warn him. Then the videos were shown to human participants and these were asked to point what seemed to be the most intelligent approach. The Theory of Mind approach was considered the most intelligent in both scenarios. They propose as a future step taking into consideration person-specific probabilities.

3.5 PsychSim

Marsella, S. C. et al. developed PsychSim [23][24], a multi-agent social simulation tool for modeling human behaviour. With this tool, a user can easily create a social scenario where diverse agents can interact and communicate among them. It allows agents to have their own preferences, relationships like friendship, authority or hostility among others, and finally private beliefs and mental models about others, or in other words, a Theory of Mind.

The simulation tool generates behaviour for the agents and provides the user with explanations for the results regarding the agent’s beliefs and preferences.

The agent’s model of the world is composed of (1) a state, with objective facts about the world, some even hidden from the agent itself, it is represented by a vector, with components valued in the range [-1,1]; (2) a set of actions, that the agent can use to change the world, an action consists of an action type, the agent performing the action and optionally an agent that is the target of the action, for example “laugh(bystander, victim)”; (3) world dynamics, that capture possible uncertain effects of actions on the following state.

Every agent has its preferences, which function as reward functions. There are two types of sub-goals (1) to minimize/maximize a feature of the agent that corresponds to a negative/positive reward proportional to the value of the given state feature; (2) to minimize/maximize an action that corresponds to a negative/positive reward proportional to the number of times that action is performed. The overall preferences and the priority among them, can be represented as a vector of weights. The product of this vector with a state vector tells us the degree of satisfaction an agent gets from the world, that is represented by that state vector.

Since human behaviour does not conform to the optimality of perfect rationality, for the beliefs about others, PsychSim modifies partially observable Markov decision problem’s algorithms in a psychologically motivated way to capture the pretended human-like behaviour.

The beliefs about others come as nested beliefs, in the sense that the agent’s belief model follows a recursive structure. For their scenario, Marsella, S. C. et al. decided a 2-level nesting was enough to generate the craved behaviour, without compromising computational efficiency. The agent’s beliefs may be about its view of the world, its beliefs of other’s beliefs, its view of other agents’ preferences, or even its view of itself. Agent’s beliefs about world dynamics will influence the way it updates its beliefs.

Agents also have policies of behaviour, a policy is the function that leads to the selection of an action or message, based on the agent’s beliefs. The policies are modelled as bounded lookahead procedures, that seek to maximize reward by simulating the effects of the selected action/message. By varying the lookahead of the policy for different agents, it is possible for them to show different degrees of reactive
vs. deliberative behaviour in their reasoning.

The problem with policies’ lookahead applied within nested models is the computational complexity, that would become impractical as the number of agents grew. To solve this problem they resorted to simplified stereotypes of the richer lookahead behaviour models of the agents. This makes the agents perform more accurately and more efficiently.

An agent may try to influence the beliefs of another with messages. A message has four components: source, recipients, subject and content. It can refer to beliefs, preferences, policies, or other aspects of other agents.

There are four key influence factors considered in their approach: (1) consistency, people try to be consistent, avoiding cognitive disparity between beliefs and behaviours; (2) self-interest, people’s reasoning and how deeply they analyze information are biased by self-interest; (3) speaker’s self-interest, if the sender of a message gains much from the recipient believing in it, there is more tendency for criticism and for influence to fail; (4) trust, likability, affinity of the recipient for the source of the message, all impact the influence on the recipient.

These factors are modelled by three mechanisms in the simulation: consistency, self-interest and bias. Consistency is the result of evaluating how much a potential belief agreed with previous observations. They use a bayesian definition of consistency based on the relative likelihood of past observations given the two sets of beliefs, current beliefs with or without believing the message. Self-interest evaluates the same two sets of beliefs, but instead of looking into the past, it looks into the future. Finally, bias factors show views of the sender that influence receiver’s acceptance/rejection of the message, they treat support and trust as this bias. Based on past interactions, agents compute their trust and support levels. The more messages an agent accepts from another, the more it trusts that agent. An agent support for another increases (decreases) when the other chooses an action with a high (low) reward, regarding the preferences of the first. An agent, upon receiving any kind of information (message or observation), must consider all of these factors in making the decision to accept that information and change its beliefs.

Since agents “know” their actions may change the beliefs of other agents, they are provided with a potential incentive to deceive. An agent may believe that with lying, other agents will act in a way that will benefit him.

The model was tested with a school bully scenario to study bullies’ behaviour considering behaviours, beliefs and properties of the victims, professors and other students. Instead of simply predicting the success of a strategy, PsychSim tells what preferences and beliefs of the bully will influence such success, and makes it possible to explore different models and understand the behaviour that results from changing psychology configurations of agents.

### 3.6 Study on Reasoning about Others

Felli et al. came up with a way to represent heterogeneous agent models, this is, with different mental abilities and holding stereotypical characteristics as part of a social group. Their goal was to support expressive, traceable and collaborative human-agent interactions[25].

For this model, agents needed to be able to represent and reason about their common ground, including beliefs and stereotypes. So they propose an approach to represent the beliefs and the model the agent has of the world and others, including nested beliefs, so that it is possible to create strategies to achieve goals. Combining this way temporal and belief projection to predict others’ future decisions.

The approach supports two types of reasoning about others: (1) stereotypical reasoning, in which the agent reasons about others using simple social rules; (2) empathic reasoning: where the agent casts
himself into another agent's mind and reasons as it was the other agent.

Instead of considering the beliefs that an omniscient observer attributes to the agents, this paper uses a local perspective of the agent in terms of its understanding of the world and of others. Making the agent not a mere executor of previously synthesized group strategies. So the agent is not limited to its internal representation and inference mechanisms, since he can use different representations and inference mechanisms for other agents.

An agent model is composed of a mental model and a set of actions with their action plausibility functions, that given a belief base tells if the action is plausible. An agent can assign this models to itself but also to others.

The mental model for an agent has (1) a set of possible belief bases, that denote its beliefs; (2) a set of axioms (or rules) to reason about the belief bases; (3) a projection, which projects a belief base to another belief base that contains only the relevant part of the original (the beliefs of the agent).

The agent models can be used in two completely different ways. One way is characterizing the individual description and understanding. The other focus more on a role of the agent, or its function in the social context, which is like the stereotypical reasoning of humans.

Ascribed mental states are used to reason as others. A mental state ascribed by an agent to another is composed by the beliefs that, according to the first agent, are held by the second. With this an agent can cast itself into the other and reason as the other would.

To reason about others they present stereotypes. Stereotypes are rules that, given a mental model for an agent, reason about the beliefs of another agent. Reasoning with stereotypes or reasoning with ascribed mental states can lead to different results.

The mental states are updated, or suffer a belief expansion, by the application of a deductive process. A new belief base is originated by a derivation from the original belief base. This derivation is a finite sequence of deductive steps, that can be formulas already in the belief base or the result of the application of a rule from the agent's mental model. Derivations' length can be bounded, which makes the agents more realistic, instead of ideal ones that are logically omniscients.

3.7 Summary

The study held by Meyer et al. introduces us to two different approaches on Theory of Mind and their implementation. It explains not only the theory behind each of them, but also the details in the implementation and presents conclusions on why we should pick one instead of the other. It also shows us results of agents with Theory of Mind compared to the results from agents without it, which we will also test in our project and from which we can take some ideas.
With a bigger focus on using Theory of Mind on producing deception, rather than detecting it (victims with less ToM levels than werewolves), Theory of Mind on Deception presents us with a complete and well explained agent model based on Baron-Cohen's approach, with characteristics we want on our own agent. Their decision on using the two level of ToM and not more for a better efficiency is something we will take into account when implementing our agent. We can claim that being this a MAS (multi-agent system), instead of a human-robot system, it is less unpredictable than ours, which will have to deal with real human behaviour from the other players.

Although is not as technical as other works here, Theory of Mind in Robotics focus on the perceptual part of Theory of Mind. Scassellati takes the best of both Leslie and Baron-Cohen’s approaches into making a complex perceptual model. Even though his systems may be a little bit outdated hardware-wise, his approach shows an interesting way to use them. In our project most perceptions will be simple instructions, although the introduction of gaze detection may be beneficial for our agent's lie detecting skills.

Although Theory of Mind in Human-Robot Teams focus on, as the name says, teams and in Coup there are no teams, it is still of great importance to deal with behaviour that differs from expected. Since the goal of each player is the same and smaller goals related to the characters are kind of deductible, we will also have to figure out unexpected actions (according to those goals) from other players and know how to deal with it but of course in our case, the agent will not be able to just ask the other players.

PsychSim provides a great alternative to our social simulation tool. Not only is it focused on Theory of Mind, but it also provides a way to easily create agents with a more human like behaviour and changeable environments to test them. Once again the described scenario is based on a MAS, not covering so much unpredictability as we expect our agent to find on his games against humans.

The study on Reasoning about Others presents two types of reasoning that are of interest to our agent. The empathic reasoning because it is necessary in Theory of Mind and the Stereotype Reasoning because there is a whole part of our agent's reasoning that will be made from stereotypes. We can find COUP's stereotypes in certain plays that, being so common, give away a character from the player that does them.
Chapter 4

Preliminary User Tests

In order to better understand the behaviour of humans while playing COUP in order to reproduce it in our agent, preliminary tests were held at GAIPS[1] where humans played COUP games against each other.

There were 14 participants and the tests were held in four sessions, three with four participants and one with only three. One of the players played twice. At least two games were played in each of the sessions.

The material used for this experience consisted in 1 COUP game set, 2 cameras, 4 microphones, 2 iPads and a hidden webcam.

The games were held on a round table with four chairs, one for each payer. The two cameras were set in a way to see every player plus the game. Each player was given a microphone that he attached to his shirt and the captured audio was later synchronized with the video. A webcam was placed next to one of the cameras in order to have real time feedback, the participants did not know they were being observed because we were afraid this would influence their behaviour.

4.1 The Session

Before the start of each session, the participants were asked to fill consent forms for the recording of video and audio for later analysis and publishing of results. They also filled some surveys in order to better understand their personality traits.

It is important to point out that the participants had little to no experience with COUP, for most of them, this was the first time playing it.

In the beginning of the session, the rules of the game were explained with a couple of demonstrations of possible plays, to make sure the rules were understood, a supervised game was held were any final doubts were answered. At the end of the supervised game the players were left alone in the room playing for approximately 30 minutes.

In order to know when a player was lying when later analysing the recording (since the camera did not capture their cards), they were asked to put in an iPad their two cards at the beginning of the game and update it every time their cards changed.

To finish the session, the participants were interviewed, with questions concerning their strategies, when and why they lied, what they had done to detect other players lies and which player did they suspect the most during the games.

4.2 Results

Through the iPad card record and the analysis of the recorded video and audio we could detect and analyse the lies participants said throughout the games. In total they lied 52 times. Table 4.1 shows the total times participants lied about having each of the characters. We can see not all the characters were lied about the same number of times.

<table>
<thead>
<tr>
<th>Characters</th>
<th>Lies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>27 (52%)</td>
</tr>
<tr>
<td>Assassin</td>
<td>11 (21%)</td>
</tr>
<tr>
<td>Captain</td>
<td>10 (19%)</td>
</tr>
<tr>
<td>Ambassador</td>
<td>2 (4%)</td>
</tr>
<tr>
<td>Contessa</td>
<td>2 (4%)</td>
</tr>
<tr>
<td>Total</td>
<td>52 (100%)</td>
</tr>
</tbody>
</table>

Table 4.1: Number of times participants lied about having a character

The character participants lied more about was by far the Duke, during the interview they admitted that lying about having the duke on the beginning of the game gave them the advantage of using Tax and collect three coins without so much suspicion because all players cards were still a mystery. Another way players lied to get more coins was impersonating the Captain, a successful Steal would not only leave them with two more coins, but also another player with two less. Even though the outcome of stealing is great, the fact that it can be countered by two characters made the player use it less, and therefore lie less about it compared to the Tax. In order to do a more secure bluff, some players faked having a Captain while having an Ambassador, this way they could block stealing without worrying about challenges, decreasing the risk of being caught to when using the Steal action. The Assassin was the second most lied about card, since doubting it is extremely risky (a failed challenge may result in two less cards, which dictate the end of the game for a player). But sometimes this was the only option left to the player due to lack of coins for coup and knowing the next turn would be his last one. Players did not lie much about having the Contessa, which as we will see ahead is good, since players tend to doubt Contessa’s counteraction a lot. The two times they lied about having it, they ended up being challenged. Players did not lie much about having the Ambassador, but since “Exchange” was not used that much, this low count is not surprising.

<table>
<thead>
<tr>
<th>Actions and Counteractions</th>
<th>Challenges</th>
<th>Succeed Challenges</th>
<th>% Succeed Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tax (Duke)</td>
<td>10</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>Assassinate (Assassin)</td>
<td>13</td>
<td>2</td>
<td>15,4%</td>
</tr>
<tr>
<td>Steal (Captain)</td>
<td>5</td>
<td>2</td>
<td>40%</td>
</tr>
<tr>
<td>Exchange (Ambassador)</td>
<td>1</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>Blocks Foreign Aid (Duke)</td>
<td>7</td>
<td>2</td>
<td>28,6%</td>
</tr>
<tr>
<td>Blocks Stealing (Capt. or Amb.)</td>
<td>8</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Blocks Assassination (Contessa)</td>
<td>21</td>
<td>2</td>
<td>14,3%</td>
</tr>
<tr>
<td>Total</td>
<td>65</td>
<td>12</td>
<td>18,5%</td>
</tr>
</tbody>
</table>

Table 4.2: Times players challenged each other and times they were right

Another thing worth analysing was the challenges made by the players. By looking at Table 4.2 we can see that just as they lied more about some characters, participants also doubted some actions and counteractions more than others.

The counteraction of the Contessa, “Blocks Assassination”, was the most challenged. It was also the second least successful challenge since only 14,3% of the times it occurred, the challenged player...
did not have the card. After the assassination is blocked, the player loses the 3 coins he payed for the assassination, so since he is already losing coins, this will leave him with a disadvantage and it may be the reason why the block is so challenged. Looking at the results we conclude that challenging this counteraction is risky, especially considering it is one of the characters players lied less about.

The Ambassador’s action “Exchange” was the less challenged, with only one of the participants challenging it. Even though it was worth it, since the challenged player did not have the card. The fact that it does not directly affect any other player, nor does it give more coins, may be the reason why this action was so less challenged.

The less effective challenge was to the “Blocks Stealing” counteraction, with 0% accuracy, since in all the 8 times players were challenged, they had either the Captain or the Ambassador. The fact that two characters have this counteraction makes it more probable that the player can indeed block the stealing and explains the low effectiveness.

Challenging the “Assassinate” action of the Assassin proved to be very risky since only 15.4% of the times it was challenged it actually paid off, which means that the other 85.6% of the times, the player doing the challenge was wrong and if it was him the target of the action, that meant the player lost the game (he lost 1 card because of the action and the other because of the failed challenge). Nevertheless, doubting the assassinate was considered to be the only option when the target of that action had only one card and could not block (either for not having the Contessa or for being a too obvious lie). This was considered by some players a “no brainer” since there was nothing to lose from it, that they would not already lose if they did not do it.

The “Steal” action was not very challenged, only 5 times. If the player has the possibility of blocking the steal, unless he is sure the other player does not have the captain (for example 3 captains faced up on the table like it happened during one of the games), he will block the steal instead of risking losing a card. It was also noticeable that players using the “Tax” action from the duke did not risk challenging the “Steal” since either way they were collecting at least one coin.

Both action and counteraction from the Duke were challenged a reasonable number of times, 10 for the “Tax” and 7 for the “Blocks Foreign Aid”. The amount of people using Tax in the first turns was sometimes too big which made it harder for players to challenge each other. This was described as one of the best plays by most of the players. Sometimes players would not do one of the two abilities of the Duke, making other players doubt they possessed the card. Being on purpose or by mistake, due to the inexperience on the game and not knowing all the actions by heart, fact is, it proved to be a great play.

When asked who they distrusted the most and the least, most of the participants said they distrusted less the passive and quiet players opposed to the more talkative players with a more aggressive stance, which they distrust more. If a player that seemed to play safe all the time suddenly changed his approach, that also raised distrust. Players that were caught lying more times would also be the focus of disbelief.

Their main strategy for detecting other players’ bluff was looking for discrepancies, for example if there are two face up characters and we have the third in my hand, if someone does an action from that character, we know that if we challenge him he’ll lose a card. Like the example given before of a player trying to steal when there were 3 faced up captains on the table.

On the interviews players admitted to try lying more at the beginning of the game, since there was less chance of getting caught (less revealed cards and beliefs about the facedown cards) and they could go on with the lie for longer. One good example was a player that started lying about having a Duke on his first play and kept on lying until the last round when he did an action from his actual character, leading the other player to challenge him and winning the game this way.
Chapter 5

Theory of Mind Agent

In this chapter we present the architecture of our agent. We start by introducing the overall model, then the main component for our work, the Theory of Mind component, and finally the deliberative component we adapted for our work.

5.1 The Architecture

Our agent architecture can be seen in Figure 5.1 and it is composed of three main components, the memory, a theory of mind component and a deliberative component.

Our agent receives perceptions, which are events from the game which can be of two types. The first are game updates, such as session start, game start, results from an action, challenge or counteraction, cards drawn, cards lost, game end and session end. These will keep the agent's memory updated at all time and will also serve to update our theory of mind. Our second type of game events, are events to make the agent play, either to pick an action, choose whether or not to challenge, choose whether or not to counteract, pick a card to lose or pick cards to keep. So these events will trigger the deliberative component into choosing the best play from the possible plays.
Each event consists of a subject, a target, an event name and finally, a set of parameters which, unlike other fields, can be any kind of value.

The Theory of Mind component keeps beliefs about other players knowledge and beliefs and, like we said before, is updated with the perceptions the agent gets, just like the memory component.

The deliberative component has as objective selecting the best play for our agent, considering all possible plays at that moment. For this it uses the information from the memory of the agent as well as from the Theory of Mind component.

5.2 Theory of Mind Component

Our Theory of Mind component has as main objective representing other players knowledge and beliefs. In Coup the knowledge we need to represent is the other players’ cards, since they are unknown to our agent. The beliefs we need to represent are what that player thinks the other players cards are. In terms of Theory of Mind levels, we need two, the first to represent a player’s cards and the second to represent what that player thinks are other players’ cards. Cards from both Theory of Mind levels are represented as a set of five probabilities, one for each role (e.g. Ambassador), which represent the probability of that player having that role card. We will firstly explain how the first level of Theory of Mind is initialized and updated, since the second level is a more complex concept that will be easier to understand once the first level is understood. To initialize the probabilities for the first level of theory of mind we use the following formula:

$$P_i(\text{Role}) = 1 - \frac{C(n', k)}{C(n, k)}$$ (5.1)

Where $n$ is the total of unknown cards to our agent (so the total number of cards in the deck plus 2 per each other player), $k$ is the number of cards in the players hand (which is always 2 in the beginning), and $n'$ is the number of cards from $n$ that are not of that Role. So the probability of the initial Role is given by the combinations of cards of that role divided by the total combinations of cards (or to be more precise, 1 minus the combinations of cards that are not of that role divided by the total combination of cards). To understand this value we can think of giving two cards from the deck (without our two cards) to the other player, if we take those two cards and ask “What is the probability of having at least one card of this role?” we get the probability of that player having that role.

So we have cards with role probabilities, but these need to change with the course of the game, depending on what happens. Every action, counteraction, challenge, or even lack of counteractions and challenges say something about what cards the players have. For example, a player not blocking a “Foreign Aid” most probably means that player does not have a “Duke” or a player challenging another’s “Tax” on the first play may be because that player has one or even two “Duke” cards. But we need a way to update this cards that makes sense, we can not just decide upon a value to take or add depending on what we see fit.

Our solution for updating the probabilities was to use data we got from previous sessions. Since Filipe Fortes[26] had done tests for his thesis using Coup we created a small database with the resulting logs from his sessions. We counted, for each role, how many times the human players did a certain action while holding that role card. Not considering for the counting roles from faced up cards. This gives us the probability of a player performing a particular action given that he has a particular role card $P(\text{Action}|\text{Role})$.

Given this information, we can now calculate the probability of an player having a card given that he
performed a particular action, using a simple Bayes Rule:

$$P(\text{Role} | \text{Action}) = \frac{P(\text{Action} | \text{Role}) P(\text{Role})}{P(\text{Action} | \text{Role}) P(\text{Role}) + P(\text{Action} | \neg \text{Role}) P(\neg \text{Role})}$$  \hspace{1cm} (5.2)$$

Both $P(\text{Role})$ and $P(\neg \text{Role})$ are obtained from the Theory of Mind probabilities.

We can calculate the values of $P(\text{Action} | \text{Role})$ and $P(\text{Action} | \neg \text{Role})$ from our small database as it follows:

$$P(\text{Action} | \text{Role}) = \frac{\text{count}(\text{Action} \cap \text{Role})}{\text{count}(\text{Role})}$$  \hspace{1cm} (5.3)$$

$$P(\text{Action} | \neg \text{Role}) = \frac{\text{count}(\text{Action} \cap \neg \text{Role})}{\text{count}(\neg \text{Role})}$$  \hspace{1cm} (5.4)$$

When we refer to Action here, it can be either an actual action from the game or a counteraction, challenge or lack of them ("No Counteraction" and "No Challenge"). When updating the Theory of Mind because of an action, the Formula 5.2 only considers actions counts from that coin range. There are four coin ranges that correspond to what actions can be used with that number of coins. The first interval ranges from 0 till 2 coins, since players can not do neither "Assassinate" (requires 3 coins) nor "Coup" (requires 7 coins), the second from 3 till 6 since players can already do "Assassinate", the third from 7 till 9 because they can do "Coup" and finally 10 or more coins since only "Coup" is available with that amount of coins.

Since there are only three cards of each character our agent needs to keep track of the cards that are known to it. For example, if our agent does "Exchange" it will know the role of 4 cards, at least until the next time another player exchanges cards with the deck, our agent will know they do not have those 4 cards (they may have others with the same roles). This is useful in case 3 of those 4 cards are the same, when that happens the probability of other players having that role needs to be set to 0. In order to do this, the Theory of Mind component keeps track of all revealed cards in two lists, the revealed cards list where it has the agent's cards and the faced up cards from other players, and the "temporarily revealed" list, where it keeps cards it has seen while doing "Exchange" and that gets reset every time another player exchanges a card with the deck.

The way we update our Theory of Mind is simpler than the Mindreading Model from Chapter 2.4, in the way that perceptions are always perceived the same way depending only on the source player. Here is how each event perceived changes our first level of Theory of Mind:

- **Game Start** - Initialize the probabilities with Formula 5.1
- **Action** - If source player is not the agent, update the probabilities using Formula 5.2
- **Challenge Result** - If source player is not the agent, update the probabilities using Formula 5.2. If "No Challenge" then do it for each player, except the agent and the player that did the action not being challenged, since any player could have challenged the action. If the challenge resulted in the target player losing a card, the probability of that player having the role that was challenged drops to 0.
- **Counter** - If source player is not the agent, update the probabilities using Formula 5.2. If "No Counter Foreign Aid" do it for every player (except for the one that did "Foreign Aid"), since any player is capable of blocking Foreign Aid.
- **Lose Card** - If source player is not the agent, lower probability of having the role of the card lost to the minimum between what the probability of that player having that card was and the probability
given by formula [5.1]. We chose to use this minimum value because the player may have 2 cards of
the same role although it is not likely. If 3 cards of the same role are revealed (even if temporarily)
drop probability of other players having that role to 0.

- Exchange Card - If source player is the agent, new card is put in the revealed list while the old card
  is put on the temporarily revealed list. If 3 cards of the same role are revealed (even if temporarily)
drop probability of other players having that role to 0. If source player is another player, empty
temporary revealed list and reset that player’s probabilities using Formula [5.1].

- Drawn Cards - If source player is the agent, put both drawn cards and faced down cards on
temporarily revealed list, empty revealed list. If 3 cards of the same role are revealed (even if
temporarily) drop probability of other players having that role to 0.

- Keep Cards - If source player is the agent, take the cards kept from the temporarily revealed list
  and add them to the revealed list. If source player is another player, empty temporary revealed list
  and reset that player’s probabilities using Formula [5.1].

5.2.1 Theory of Mind Level 2

Our Theory of Mind follows a Simulation-Theory approach in the way that our agent will consider other
players to reason the way it does. So our second level of Theory of Mind will be updated using the same
formulas as the first.

When the agent initializes the first level it has in consideration its two starting cards and their roles.
But when updating the second level, since it does not know the roles of the cards of the other player, it
assumes the three cards from that role are in the deck, thus in Formula [5.1] we have \( n' = n - 3 \).

Formula [5.2] is used the same way as in the first level, with \( P(\text{Role}) \) and \( P(\neg \text{Role}) \) coming from the
second level probabilities.

The “revealed” and “temporarily revealed” lists are not used for this level since those cards are only
revealed to the agent and not the other players. For the second level the Theory of Mind component
keeps an extra “revealed to all” list where it keeps the faced-up cards in order to update the second level
of theory of mind properly.

Here is how each event perceived changes our Theory of Mind second level:

- Game Start - Initialize the probabilities with Formula [5.1].

- Action - Update probabilities referring to source player from all other players using Formula [5.2].

- Challenge Result - Update probabilities referring to source player from all other players using
  Formula [5.2]. If “No Challenge” then do this to probabilities referring to each player, except the one
  that did the action not being challenged. If the challenge resulted in the target player losing a card,
  the probability of that player having the role that was challenged drops to 0 for all other players.

- Counter - Update probabilities referring to source player from all other players using Formula [5.2].
  If “No Counter Foreign Aid” then do this to probabilities (from all players) referring to each player,
  except the one that did “Foreign Aid”.

- Lose Card - Put lost card in revealed to all list. Lower probability on others of that player having the
  role of the card lost to the minimum between what the probability of that player having that card
  was and the probability given by formula [5.1]. If 3 cards of the same role are revealed to all drop
  probability of other players having that role to 0 in all players.

- Exchange Card - Reset that player’s referring probabilities on other players using Formula [5.1].
- Drawn Cards - Do nothing.
- Keep Cards - Reset that player's referring probabilities on other players using Formula 5.1

5.3 Deliberative Component

For the deliberative component we will use Filipe Forte's Decision Making Algorithm [26]. This Decision Making Algorithm is an adaptation of the regret minimization algorithm [27].

Since Coup is a game where some information is unavailable (the cards of the other players), a player can regret making some action once he learns its result and finds out this would have been better had he chosen another action. This algorithm goal is to minimize the regret of the agent.

The decision making algorithm takes current information about the other players directly from the Theory of Mind component, and the current state of the game, which is taken from the information processed in the Memory component of our agent. It keeps strategies from previous plays, which are composed of information about the actions from each respective play.

For each action that is available to the agent in the current state of the game, the algorithm will generate all possible outcomes for that action. The algorithm will also generate for each of these outcomes their corresponding probability (e.g. for the “Steal” action, the possible outcomes are it succeeds, it gets challenged or it gets blocked), that is obtained from the information the agent has stored in its Theory of Mind component. Note that here the probabilities of actions being challenged or blocked were set to default values that did not change throughout the game.

Having all possible outcomes and their probabilities for the available actions, the algorithm will calculate, for each outcome, its value towards winning the game (given by a utility function that calculates the potential of the player winning the game based on his potential to get coins and eliminate cards), which is then multiplied by the outcome corresponding probability. The value of the action is given by adding all of that action’s outcomes product between value and probability.

By multiplying each action's value by its probability in the previous agent’s strategy, the algorithm produces the value relatively to that previous strategy. This will be done for all past strategies.

Having the value of past strategies towards the current state of the game, the algorithm will proceed to calculate the regret of each action by summing the difference between the action’s value and a past strategy's value, multiplied by the probability of reaching the current state of the game using that past strategy. By summing all these values for each past strategy and then dividing by the number of past strategies, the algorithm is capable of effectively calculating the regret for each action.

A new strategy is then produced based on the percentage of regret that each action gives, when considering the total amount of regret of all actions summed. After having the new strategy defined, the agent will then play the action that produces the least amount of regret.

The process mentioned before is only used when a decision involves an action, challenge or counteraction. When the decision involves losing a card or changing cards, the algorithm uses a different approach.

This different approach is very similar to the one for choosing a normal action, but for losing or exchanging cards, the calculations only take into account the value of the outcomes of losing or changing certain cards.

After calculating the value of the outcome for each possible action of losing or changing cards, the algorithm selects the action that provides the most value for our agent or, in other words, that provides the biggest potential to reach a favorable end of game state.
5.3.1 Accommodating Theory of Mind

The Theory of Mind is used for two main tasks in Coup, detecting lies and lying. The lie detection is done with the first level of Theory of Mind probabilities, if a player plays an action from a role and the probability of having that role is low, the “Challenge” will be considered the best play by the Decision Making Algorithm. Both the first and second level will help the player choosing its actions as we will see ahead, but mostly the second level will be used for lying, since it will give us insight to whether or not the other player will believe our agent’s action, according to the cards it believes our agent to have.

A positive aspect of the Deliberative Component we used is that it was easy to integrate the Theory of Mind probabilities, since it used some default values before that we replaced them by our probabilities.

But making sure the algorithm used our probabilities was not enough, we had to make rules when choosing the best action to ensure the good functioning of the Theory of Mind.

We lowered the threshold for doubting “Exchange” since it is not an action players lie much about. So our agent will only doubt challenges when the probability of the other player having the card drops below 10%.

We also made sure that when the agent had the card to block either “Steal” or “Assassinate” it would not challenge those actions unless the probability of the other player having the card was really low (below 10%).

Since most players lie about having a “Duke”, specially in the beginning of the game, the initialized Theory of Mind values would not be enough to catch these lies, since our agent would only challenge “Tax” in the beginning of a game if it had 2 “Duke” cards. We wanted our agent to risk it sometimes, but not too much, so we decided that if it has one “Duke” card it will doubt an early “Tax” (on the other player’ first turn) 50% of the times.

Another very important rule we made was for the agent not to challenge any action if the game can be won on the next turn (other player only has 1 card and the agent has 7 or more coins). With the exception of an "Assassinate" that would make the player lose or a “Steal” that would leave the player with not enough coins to win the game, in these cases if the agent has the card it takes to block it does not challenge, if not we let the decision making algorithm decide whether or not it wants to challenge or lie about being able to block.

One thing we had to implement was the probability of the agent being challenged and countered, since it was initialized with default values and left like that throughout the game. These are important to ascertain when it is safe to lie and when it is not, and also to avoid doing action that will most likely be blocked. To generate and update this probabilities we used information from the two levels of Theory of Mind. As a result we have the following functions:

- \( P(\text{BlockForeignAid}) = \text{OtherPlayer}.P(\text{Duke}) \)
- \( P(\text{BlockAssassinate}) = \text{OtherPlayer}.P(\text{Contessa}) \)
- \( P(\text{BlockSteal}) = \max(\text{OtherPlayer}.P(\text{Ambassador}), \text{OtherPlayer}.P(\text{Captain})) \)
- \( P(\text{ChallengeTax}) = 1 - \text{OtherPlayer}.\text{Other}.P(\text{Duke}) \)
- \( P(\text{ChallengeAssassinate}) = \frac{1 - \text{OtherPlayer}.\text{Other}.P(\text{Assassin}) + (1 - P(\text{BlockAssassinate}))}{2} \)
- \( P(\text{ChallengeSteal}) = 1 - \text{OtherPlayer}.\text{Other}.P(\text{Ambassador}) \)
- \( P(\text{ChallengeSteal}) = \frac{1 - \text{OtherPlayer}.\text{Other}.P(\text{Captain}) + (1 - P(\text{BlockSteal}))}{2} \)
- \( P(\text{ChallengeBlockForeignAid}) = 1 - \text{OtherPlayer}.\text{Other}.P(\text{Duke}) \)
- \( P(\text{ChallengeBlockAssassinate}) = 1 - \text{OtherPlayer}.\text{Other}.P(\text{Contessa}) \)
- \( P(\text{Challenge Block Steal}) = 1 - \max(\text{Other Player Other} P(\text{Amb.}), \text{Other Player Other} P(\text{Captain})) \)

OtherPlayer.P(Role) represents what our agent thinks the other player's cards probabilities are. OtherPlayer.Other.P(Role) represents the second level of Theory of Mind, or what we think the other player thinks our cards are.

Besides this, whenever the other player blocks an action, the probability of block for that action will be 1.0 until its cards suffer some change. Note that the algorithm may still choose that action if for some reason the probability of the player having the role capable of doing the block drops too much (in order to after challenge the block).
Chapter 6

EMYS, the Coup Player

In this chapter we explain how our player works, its components and their integration. We also describe how the human-robot interaction is performed.

6.1 Overall System Architecture

Our complete system is composed by the agent from Chapter 5, a robot (EMYS), a behaviour planner (Skene), a text to speech component and of course, the coup game itself. All of these components need to communicate with each other throughout the game, be it to give and receive instructions on what to do and what was done in the game, or to communicate with the human player. In order to connect all of these components we used Thalamus[28], a high-level integration framework that makes it possible to divide an interactive character into several components, both virtual and physical. This means that each component can perform its role and cooperate with other components through the exchange of messages on thalamus. Thus we just need a client for each component, which will subscribe to the events it wants to receive from and send to Thalamus. Figure 6.1 shows all components connected via thalamus. Even though they are all connected, not all components will communicate between themselves, only in the subscribed channels.

![Figure 6.1: Overall System Architecture](image)

The agent’s implementation was done on FAtiMA[29], an agent architecture with planning capabilities, capable of influencing agents’ behaviour with the use of emotions and personality. FAtiMA’s core is a template with generic functions that describes how the Agent Architecture works. By itself, the core is insufficient for the agent to do anything. The core’s architecture is shown in Figure 6.2. The agent receives perceptions from the world that are used to update its memory, where its knowledge is stored,
and the affective state, which stores emotions and moods. Both the affective state and memory can be used to influence the selection of an action that will act upon the world.

![Figure 6.2: FAiMA core architecture](image)

We made use of FAiMA’s modularity to add important components to the core that best suit this project. Even though we did not need any of the appraisal and emotions FAiMA offered us, using FAiMA made it easier to implement the agent, thanks to already implemented ways of dealing with events.

### 6.2 EMYS Robot and Speech Component

The robot we used is called EMYS (EMotive headY System) [30] and was designed and built within the EU FP7 LIREC project [1]. EMYS is a mechanoid robot head conceived for human-robot interaction experiments like ours and can be seen in Figure 6.3. It consists of three discs equipped with a pair of movable eyes with eyelids, mounted on a movable neck. We named our player EMYS after the robot because people will be physically playing against the robot.

EMYS is able to act upon the environment and also to perceive it. To act, it uses its previously described components together with a speaker. These allow it to make facial expressions in order to express emotions, establish and maintain eye-contact with humans, and play synthesised or prerecorded speech, giving it a voice. To perceive, EMYS has a colour CMOS camera Logitech Sphere AF and can be equipped with a Kinect sensor and a lapel microphone. Enabling it to visually perceive the environment, eyetrack objects and humans, pay attention, and recognize speech.

These capabilities made this robot head a great host for our agent, making it possible for the agent to interact with the human players. For our work, we only want it to interact in two ways, by speech and turning his attention towards different things. We think that using the other capabilities besides greatly increasing our agent’s complexity, would not be very helpful to achieve this work’s goal.

To control EMYS we use a semi-autonomous behaviour planner capable of semi-automated behaviour called Skene [28]. For instance, to generate speech-based behaviour with skene, one just needs to write a set of utterances on a spreadsheet, divided by categories and subcategories (with the format Category:SUBCATEGORY). Once our agent tells skene to perform an utterance of a certain category

---

and subcategory, skene will output BML (Behaviour Markup Language) and non-BML actions (such as sounds or application commands). Which will be sent to the EMYS component to make the robotic lip sync and to the Speech Client component to reproduce sound according to the utterance. The EMYS component will also receive instructions to look to a certain coordinate from Skene, this way we can tell EMYS to look to the player, to its cards and to the player's cards.

For the agent to choose the right sentence according to what is happening in the game we needed to implement a manager for the utterances on the agent side. FAtiMA already had actions of type "speech-act" but we did not want to have different types of actions. Whenever EMYS would need to make an action in the game, it would have to communicate the action both to the game and to the human player, producing two actions, which would result in one overriding the other. So, to avoid this problem, we decided to create a new module for FAtiMA called utterance manager. An updated version of the agent with the utterance manager module can be seen in Figure 6.4.

![Figure 6.4: Agent architecture with the utterance manager module and player information](image)
The utterance manager has two main functions: producing an utterance according to the action chosen by the deliberative component and producing an utterance according to perceptions from the game. In other words, it creates utterances for actions and reactions. An utterance has a category and a subcategory, all categories and subcategories as well as an explanation for when they are used can be seen in Table A.1 of Appendix A so if the deliberative component outputs the action “Steal” as the best action, the utterance selected will be of category “Action” and subcategory “STEAL.”

In order to keep coherence the utterance manager takes into account the action sequence. For example, instead of saying “I don’t doubt it” (NoChallenge category) followed by a “I block” (Counter category), EMYS will say “I don’t doubt it” followed by a “but I block” (ButCounter category), in order to create a more natural speech.

EMYS comments to game events take into account results from actions, challenges and counteractions, not only its own but also the other player’s results. It also has something different to say at the end of a game according to the winner, the game’s length and the number of cards lost by EMYS. A game in which EMYS wins while keeping both its cards faced-down will result in an utterance of the type “GameEnd:EASY”, while a game that only lasts 4 rounds or less will be a “GameEnd:QUICK”.

To generate new utterances when the player plays more than one session, the utterance component makes use of information the agent stores about the other player (Figure 6.4 Player Information) such as actions, challenges results, number of no challenges and previous sessions results. This allows the agent to trigger special comments when the other player uses an action more often than any other throughout the games. For instance, in case a player uses “Steal” too often it will trigger a comment like “You’re always using steal, should I call the police?”. Another type of comment that can be triggered is when a player doubts EMYS too much, it will say something like “Why are you always doubting me?”. This way we hoped to increase the way the users perceive EMYS as capable of social interaction.

Besides saying its actions and commenting on certain game events, EMYS is also capable of reminding players who take more time to play or did not understand what EMYS did. It does this if it does not receive any new event for a certain period of time, it sees what the user needs to be reminded of depending on the game phase and does so. So if a player is distracted while EMYS does a Tax, after some time EMYS will say “Don’t forget I did Tax”.

6.3 Digital Tabletop and Coup Unity Game

Since EMYS has no hands we used a digital tabletop running a Coup game, to allow human players and robot to play together.

A MultiTaction Ultra Thin Bezel Display was used as tabletop for the game. This 55” interactive multiuser LCD has an ultra-fast response time and is able to track unlimited touch points, including hands, fingers, fingertips, 2D markers and real-life objects.

We used the Coup game developed for the tabletop by Filipe Fortes for his study. The game was developed in unity and communicates with Thalamus to either get actions from the virtual agent or to update it. Due to possible network delays and events received in a wrong order, the events are queued and then tested to see if they are valid in the current game phase. Everytime something changes the game state, may it be because of another player’s actions or the result of the agent’s own action, unity sends the update through Thalamus.

The user communicates with the game through the tabletop. The game interface can be seen in Figure 6.4. Every player, EMYS included, has in front of him all the information it needs to play. This information includes their cards, that are hidden by default, a button that shows or hides the cards,
depending on their current visibility, a number that represents the player’s current number of coins and finally, a list of actions. The actions list is just like the original cheat sheet, it not only serves as buttons for the players to do the actions, but also provides all the information they need in terms of actions, counteractions and which characters may be needed to perform them. Using any action that requires another player as a target ("Coup", "Assassinate" or "Steal") will create a new list of players to target, so the player can choose who to use his action on. All actions and counteractions that can be challenged, will prompt all other players, that are still playing, if they want to challenge the action, the same will also be done when an action can be countered, asking the players that are allowed to counter it, if they want to do so. These messages appear with a “Yes” and a “No” buttons for the player to click on. An example of this is shown in Figure 6.6, where a player has to decide whether to challenge a "Steal" or not.

Based on users’ feedback taken from Filipe’s tests we identified a list of problems with the interface
and interaction:

- **EMYS was ignored** - We noticed on the videos from the tests that players did not look much to EMYS. Mostly because they would always be looking at the interface trying to figure out what was happening.

- **Too much text** - The interface had too much text, which players had to read in order to understand what was going on, this did not feel natural and sometimes it got confusing for the players.

- **Too static** - Sometimes players did not really know what happened within the game, or if it was their turn or not. Due to the changes on the interface being mostly text-wise.

- **Upside down text** - The EMYS's number of coins text was upside down for the players, making it harder for them to read it.

- **Lack of color** - The interface lacked color, it was "too white" in its whole, making it look a bit monotone and giving the illusion that it was a prototype of the final version.

- **Action Buttons** - There was too much text on the action buttons and most of the time it was hard to perceive them as buttons. The fact that those buttons were all together with no gap separating them made player click on the wrong button, ending up doing actions different from what they pretended.

- **Yes and No buttons** - Another problem with the buttons was was that the delay between some of the phases was non existent, which in turn made some participants choose a wrong action. An example of this is when there was an action that can be blocked, if a player pressed the "Yes" button almost at the same time as another player, the one to press it secondly ended up challenging the block instead of making the block himself. This because the counter phase would automatically change to the challenge counter phase and the buttons would remain "Yes" or "No".

In order to fix these problems and make the interface more appealing and user friendly we did some changes. Figure 6.7 shows the new interface.

The answer for both EMYS being ignored and having too much text was taking all the text (except the one from from buttons) from the interface. This way the player would have to pay more attention to EMYS during the game, making it more natural. In a normal game of Coup between humans the players just say which action they want to do and in case the other player was not paying attention they repeat it. So that is what we did, like we said before EMYS now reminds the player of the action he did if this takes too long to play.

In order to make the interface less static and for the player to better understand when to act, every button that does not need to be showing at a given time will be hidden until it does. So when it is the player's turn to act, the buttons to do so will appear, making it clear to the player what he needs to do. We can see from Figure 6.7 that it is the player on the right turn to play because it is the only player with that option.

Since the number of coins text was upside down, we decided to change it to actual coins (yellow circles on Figure 6.7), this way the player just needs to count the number of coins the other player has just like the real game experience. These coins make the interface look better, but it was not the only thing we did in order to add more color to it. We put a metal-like background (since the original game happens in a futuristic scenario) and changed the "Show/Hide cards" button making it a green "Show cards" button when the cards are hidden and a red "Hide Cards" when they are visible, this way we also hope the player will not forget to always hide its cards. On Figure 5.8 we can see the changes clicking on the "Show cards" button does.
To make the game experience more natural we got inspired for this next feature from a cheat sheet that comes with the Coup original board game. All the detailed information about characters, actions and counteractions present in that sheet was already in the actions button but it was clearly too much information to have on simple buttons. So we decided to create a help button symbolized by a circle with a “?” in the middle that while pressed shows a copy of that sheet. When the player lets go of the button the sheet disappears, we did this in order for the player not to be looking at it all the time, only when needed. On Figure 6.9 we can see the “sheet” that pops up when we hold the help button.

Since we now have all the information in the help button, we took all non-essential text from the action buttons, leaving just the name of the action for the user to select. We also spaced them a little bit in order to avoid miss-clicks. These can be seen in Figure 6.7 next to the cards of the player on the right.

Since we did not have any more text, the Yes and No buttons stopped making sense, unless we would make EMYS say “Do you block?” and “Do you challenge?” everytime, which would not sound natural. So we decided to change the buttons to “I Doubt” and “I believe” for the challenge phase and “I block” and “I don’t block” for the counteraction phase. This not only helps turn the game more natural but also helps differentiating blocking from challenging. Figure 6.10 shows the challenge buttons and

Figure 6.7: Coup game’s improved interface

Figure 6.8: Cards showing with “Hide button” (Esconder)
Finally, we delayed the change of phases in order to avoid the buttons changing right before a click from a player.

We also took out the target list in case of the game being played by only two players, since it made no sense having to click a second button.
Chapter 7

Evaluation

In order to test our social robot playing the game of Coup we conducted user studies, where users played Coup against our robot. Our objective here was to test our hypothesis: an agent equipped with Theory of Mind shall detect other players' lies more often while playing the game COUP against a human than one without Theory of Mind. Besides testing the hypothesis we wanted also to find out if our Theory of Mind component had influence on how EMYS plays the game of Coup and on humans' perceptions of EMYS' skills. In this chapter we will explain how the tests were conducted, present the results and our analysis of the latest.

7.1 Sample

A total of 30 users took part of this study, where 22 were male and 8 female, with ages ranging from 22 to 33 (M=24.07; SD=2.506). Participants were randomly allocated to one of two different conditions: the Theory of Mind condition, where our agent was using the Theory of Mind module from Chapter 5 and a no Theory of Mind condition, where the agent would not use the module. In both conditions, all the other modules were the same (except of course for the Theory of Mind influence on the deliberative component). So the agent without Theory of Mind does not keep any information on the other player’s cards or beliefs about EMYS cards, making it riskier for this agent to challenge the other player’s actions. The participants were not made aware of the different types of the agent so their initial perception of the agent did not differ between the two conditions. 44.8% of the participants had already played Coup while 55.2% had not. At the end, only 29 of the participants were used for the sample, since one of the players was excluded from the analyse due to showing an outlier behaviour in almost every question in comparison with the remaining sample.

All participants signed a consent form so that all the information they provided could be used in this study.

7.2 Measurements

We did two types of measurements, a questionnaire, to get how users perceived the two agents, and we kept logs from all plays done in the games. The logs contained the following information: number of player, action, role of card number one, role of card number two, number of coins and two booleans, one for each card, with true or false value depending on whether or not the card is still in play. The questionnaire is described ahead.
In order to ascertain the robot’s competence we used Godspeed’s dimension of perceived intelligence, with a semantic differential of 5 points\cite{32}.

Fifteen other questions of the Likert-type scale with 5 choosing points were created, based on the competitive context and on the game’s objective, covering this way:

Theory of Mind:

- Q4 “I think that EMYS perceived what my cards were”
- Q5 “I think that EMYS perceived when I lied”
- Q7 “I think that EMYS could understand my game strategy”
- Q14 “EMYS could anticipate my decisions”

Robot’s Performance:

- Q1 “I think that EMYS was a strong opponent throughout the game”
- Q2 “I think that EMys was capable of lying well throughout the game”
- Q3 “I think that EMYS was capable of deceiving me throughout the game”
- Q6 “I think that EMYS played in a competitive way”
- Q8 “I think that EMYS played well”
- Q9 “I think that EMYS lied as well as a human”
- Q11 “It was challenging to play with EMYS”
- Q12 “EMYS plays made me change my tactic during the game”
- Q13 “EMYS was easily deceived by me”

Enjoyment:

- Q10 “I liked playing the game with EMYS”
- Q15 “I was bored with the course of the game”

Answers ranged from 1 - “Totally Disagree” to 5 - “Totally Agree”.

### 7.3 Methodology and Procedures

Each player had to play three sessions of five games each, on different days. Our objective with playing multiple sessions was to take away the downside of inexperience. Since Coup is a complex game, people with less experience would have more trouble perceiving the robot’s skills. The reason why we held individual games instead of three humans playing with EMYS like in Fortes’ tests\cite{28} was because human players tend to eliminate EMYS quite early in the game or just not paying too much attention to it throughout the game.

On the first session, upon arrival to the experiment room, the participant would answer to the first part of the questionnaire, composed of demographic questions and questions 1, 2, 3 and 7 before interacting with the robot (in order to ascertain their expectations). Then if the user did not know the game beforehand, we would start by explaining its rules with a copy of the traditional Coup board game. Once the player knew the rules we would then explain how to play the game on the tabletop. We would
do this by supervising the first game, in order to make sure the user understood correctly both the rules and also our interface, to avoid mistakes during the tests. The remaining games were done with no on-site supervision. At the end of the five games, the participant would answer to the second part of the questionnaire with the same four questions from the first time (now on past tense to see the difference between satisfaction and expectation) and the remaining questions.

On the second session participants just played the five games, unsupervised, without filling any questionnaire.

Finally on the third session, participants played the five games with the robot and, at the end, filled the same questionnaire they filled at the end of the first session.

While playing the game there was a hidden camera filming the interaction of the players with our robot in order to help evaluating our results. The players were not made aware of this until the end of the last session, where they agreed to the use of the footage for the study.

As a thanks for their participation, every participant received three cinema tickets.

7.4 Questionnaire Results

The data analysis from the questionnaires was done taking into account the observed population’s normality, whenever this was observed the statistic test used was t-Student for independent samples. In cases where normality was not observed we used Mann Whitney-U as a non parametric test. To ascertain the interaction between the time effect (sessions) and the conditions we ran a Mixed Anova, since we compared data from repeated measurements with independent samples’ data. There was no significant changes between the time (the session) and the condition in which the participants were.

7.4.1 Perceived Intelligence

From Godspeed’s dimension of perceived intelligence we got that participants considered EMYS intelligent regardless of conditions or sessions as can be seen in Figure 7.1.
7.4.2 Theory of Mind

The four questions, Q4, Q5, Q7 and Q14, had as an objective to see if humans noticed the presence of Theory of Mind on the condition where it was present. The results obtained from our participants were not conclusive, being that none of the results had significant differences between conditions.

The participants seemed to think EMYS did not know what their cards were, with results from both conditions with scores close to the neutral value. We found it strange that both conditions got similar results, since the condition with no Theory of Mind did not catch much lies (as we will see in the discussion at the end of this chapter) from the players. Although the neutral values may indicate players just did not know if EMYS knew or not what cards they had.

In terms of EMYS knowing when the participants lied, the results were again inconclusive between conditions. With the results from third season (U=64.5, p=.067) implying that there might be a slight difference, just not a significant one. We believe that with more participants we would have a significant difference between the two conditions, with the Theory of Mind condition getting a better score. Figure 7.2 shows the means for both conditions on third session.

We now believe that the question, "I think that EMYS perceived when I lied", should have been preceded by "Did you lie during the game?", because players who did not lie could not possible know if EMYS knew when they lied. So if there were players that did not lie, they probably put 3 on this question.

![Figure 7.2: 3rd session means from question 5 "I think that EMYS perceived when I lied".](image)

Participants also thought that EMYS was not capable of neither understand their strategies, nor anticipate their decisions, with results being very neutral (around 3). We believe that the complexity of Coup plays an important role in these results since both strategies and playing in anticipation are not something a less experienced player pays attention during the game. Again a question that may be hard for less experienced participants, "I think that EMYS could understand my game strategy", if a participant did not have a particular strategy he probably did not know how to answer this question. Leading to more neutral results.

7.4.3 Robot’s Performance

With this set of questions we wanted to know what the users thought of our Robot’s performance in the game.

Overall, participants in both conditions perceived EMYS as a strong opponent, that played well and
was capable of lying well throughout the games. They also considered playing with EMYS to be challenging and that its plays would make them change tactics during the sessions. There were no significant differences between the conditions in any of these aspects on either of the sessions.

Although participants in both conditions seem to have felt that EMYS was able to deceive them during the game, results from participants in the Theory of Mind condition show a significant difference (U=63.5, p=.05) compared to the other condition, with higher scores on the third session. In Figure 7.3 we can see the means of both conditions on the third session.

Figure 7.3: 3rd session means from question 3 "I think that EMYS was capable of deceiving me throughout the game".

In respect to competitiveness, even though in general participants seem to have felt that EMYS played in a competitive way, on the third session participants in the Theory of Mind condition perceived EMYS as more competitive than the ones in the other condition (U=63, p=.047) as can be seen in Figure 7.4.

Figure 7.4: 3rd session means from question 6 "I think that EMYS played in a competitive way".

In general participants from both conditions seemed to think EMYS was not easily deceived by them. On the third session though, results seem to show that the participants in the condition with no Theory of Mind gave higher values than those in the Theory of Mind condition, as we can observer
Even though we may be inclined to think this, we cannot say it for sure due to lack of significant differences in these results ($U=67.5$, $p=.089$).

![Figure 7.5](image)

Figure 7.5: 3rd session means from question 13 “EMYS was easily deceived by me”.

### 7.4.4 Enjoyment

With the two enjoyment questions, Q10 and Q15, we wanted to see if participants enjoyed playing with EMYS. Players in both conditions liked playing Coup with EMYS. We found significant differences between the two conditions ($U=70.5$, $p=.042$) on the results of the third session, where it shows that participants placed in the Theory of Mind condition seem to give slightly higher values to Question 10 than participants in the condition without Theory of Mind as can be seen in Figure 7.6.

![Figure 7.6](image)

Figure 7.6: 3rd session means from question 10 “I liked playing the game with EMYS”.

We think that the fact that participants seemed to perceive EMYS as more competitive in the Theory of Mind condition (Q6), may be the reason why there is this difference in liking to play with EMYS.

In general, participants in both conditions did not find playing with EMYS boring with the course of the game. Values from both the first and the third sessions were close to 1 (Totally disagree), with a
significant difference (U=70, p=.040) between conditions in the first session, the means from this session can be seen in Figure 7.7.

Figure 7.7: 1st session means from question 15 "I was bored with the course of the game".

7.5 Empirical Results

Before we start discussing the results, we will present some relevant numbers obtained from the recorded logs of Coup games.

The difference between the final number of wins, both in games and sessions, of the two conditions was not conclusive. In total, the Theory of Mind agent won 109 games (out of 225) and 19 sessions (out of 45) while the agent with no Theory of Mind won 106 games (out of 225) and 18 sessions (out of 45). So both agents won around the same percentage of games, 48% and 47%, and sessions, 42% and 40%.

In the Theory of Mind condition, EMYS lied a total of 141 times. The human players in this condition challenged EMYS a total of 225 times and caught 71 of its lies. So 50% of EMYS lies were caught by the players, who had 31% successful challenges. On the other hand participants lied a total of 151 times. EMYS challenged them 190 times and caught 72 of their lies. Making it 48% of lies caught by EMYS and 38% successful challenges.

In the condition with no Theory of Mind, EMYS lied 246 times. The human players in the condition challenged EMYS 294 times and caught 109 of its lies. EMYS lies were caught 44% of the times and the players had 37% of successful challenges. In terms of participants in this condition, they lied a total of 254 times. 4 of these lies were caught by EMYS that made a total of 13 challenges. So EMYS only caught 1% of the players’ lies, with 31% of its challenges being successful.

All of these values can be seen in Table 7.1 and Table 7.2.

The numbers seem to confirm our hypothesis: an agent equipped with Theory of Mind shall detect other players’ lies more often while playing the game COUP against a human than one without Theory of Mind. With 48% of lies caught by the agent with Theory of Mind against 1% and 38% of successful challenges against 31%.

Not only is the percentage of detected lies and successful challenges bigger compared with the no Theory of Mind condition, but one could also say it is as good as the human participants. Participants in the Theory of Mind condition caught 50% of the lies, only 2% more than our agent with Theory of Mind.
<table>
<thead>
<tr>
<th>EMYS</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lies Told</td>
<td>141</td>
</tr>
<tr>
<td>Successful Lies</td>
<td>70</td>
</tr>
<tr>
<td>% Successful Lies</td>
<td>50%</td>
</tr>
<tr>
<td>Challenges Made</td>
<td>190</td>
</tr>
<tr>
<td>Successful Challenges</td>
<td>72</td>
</tr>
<tr>
<td>% Successful Challenges</td>
<td>38%</td>
</tr>
</tbody>
</table>

Table 7.1: Lies in the Theory of Mind condition

<table>
<thead>
<tr>
<th>EMYS</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lies Told</td>
<td>246</td>
</tr>
<tr>
<td>Successful Lies</td>
<td>137</td>
</tr>
<tr>
<td>% Successful Lies</td>
<td>56%</td>
</tr>
<tr>
<td>Challenges Made</td>
<td>13</td>
</tr>
<tr>
<td>Successful Challenges</td>
<td>4</td>
</tr>
<tr>
<td>% Successful Challenges</td>
<td>31%</td>
</tr>
</tbody>
</table>

Table 7.2: Lies in the No Theory of Mind condition

and were 6% less successful in terms of challenges, with 31% successful challenges against 37% from the agent. Participants in the other category caught even less, 44% against 48% of caught lies with 37% successful challenges.

7.6 Discussion

Some relevant questions rise from these results, to which we try to answer ahead.

**Why did the agent with no Theory of Mind challenge so little times the participants?**

The agent in the No Theory of Mind condition has no idea of what cards the other player has. This being said, it does not challenge the human player often because it is too risky and the Decision Making Algorithm ends up choosing the “No Challenge” over the “Challenge”. There are only two situations in which this does not apply, when the algorithm sees there is no more chance of winning or the agent saw the three cards of that character and the other player’s cards have not changed since then. The first situation is usually when the other player blocks EMYS’ “Assassinate” and the other player will to win next turn either way.

**Why did both the agent and the participants in the No Theory of Mind condition lie more than the ones in the Theory of Mind condition?**

The reason why the agent in the No Theory of Mind condition lied more than our Theory of Mind agent was because it did not speculate over whether or not the other player would challenge its action. The Theory of Mind agent does this, avoiding this way too obvious lies. One example of this is when the agent loses a card because it did not have, for example, a “Duke” upon being challenged for doing “Tax”, the Theory of Mind agent will update the information on the other player so that it knows the other player now knows EMYS does not have a “Duke”. The other agent does not do this, so it may still make “Tax” the next turn. We believe that the participants on the same condition on the other hand, lied more because they were not challenged that much. Feeling free to lie at will.

**Why did the agent with Theory of Mind lose almost the same number of games as the one without it?**

After an analyse of all the games done by the Theory of Mind agent, we noticed that it made frequent risky challenges, losing a lot of cards this way. In 70% of the matches this agent lost, it lost at least 1 card for an unsuccessful challenge and in 15% it lost both its cards for the same reason. Even though our
agent’s successful challenges rate is higher than the participants’ one, we still think it can be improved. One thing that would do this, would be improving the reasoning in the deliberative component to assign risks for losing the second card higher than losing the first, making it more careful and only doubting when almost certain (lower probability of other having the character).

**Why did the participants in the Theory of Mind not give significant better values in the Theory of Mind questions, compared to the other condition?**

We believe that even though our Theory of Mind lead the agent on the first condition to detect as many lies as a human, it may not be that obvious for the participants what that standard is, so even if EMYS behaved well, it may have not be enough for people to perceive that it knew their cards and when they lied. The questions may be the problem, since that, even if the agent got values close to those of the humans, lie detection wise, one would not say that a person that failed around 60% of challenges actually knew what the other player cards were or when they lied.

Even though there were not significant differences between the two conditions results, on the third session, we get the impression that the second condition rates tend to be lower, at least in the questions 4 and 5 (both means are under 2.5). We find it weird that some players gave a rate of 4 to either of the questions in this condition, since the agent made only 13 challenges in total against all participants, the only reason we can think of is that their answers were acquiescence bias. Acquiescence bias, or friendliness bias, occurs when a respondent demonstrates a tendency to agree with and be positive about whatever the moderator presents.

**Why was the Theory of Mind agent perceived as a better liar than the agent with no Theory of Mind, when the latest had a slightly bigger percentage of successful lies (54% against 50%)?**

We think that the reason the Theory of Mind agent has slightly less successful lies was because games were shorter on the Theory of Mind condition (approximately 12 turns per game, compared to an average of 17 turns per game in the other condition) because the challenges done made the games end earlier. A longer game helps the player keep their lies for more turns without being challenged, since usually if a player does not challenge an action in one turn, it will believe it in the next. The fact that participants from the Theory of Mind condition were less successful with their challenges, with 69% failed challenges against 63% could be the reason why the Theory of Mind agent was perceived as a slightly better liar than the other.

### 7.7 Conclusions

With this study we were able to take some conclusions about our agent performance at playing the deception game Coup and how that performance was perceived by the participants.

We came to the conclusion that our agent is indeed capable of detecting lies better than one with no Theory of Mind, and even detect lies at the same level of humans. Thus proving that our initial hypothesis is correct.

In terms of perception from the players we did not get many significant differences, but the Theory of Mind agent was perceived as a slightly better liar, more competitive and more enjoyable to play with.
Chapter 8

Conclusion

Even though there are many areas to where theory of mind is being put to use, there are no significant studies on its use on agents lie detection skills. With this dissertation we intend to change that.

We started by providing a theoretical background focused on Deception, Lie Detection and Theory of Mind to ease the understanding of our approach. Where we also explained the game in which our agent will be interacting. We proceeded by presenting studies that related somehow to what we aspire to achieve, and gave us some insight on how to achieve it. We saw theory of mind approaches and the use of theory of mind in areas like deception, robotics and human-robot teams’ behaviour. We also analyzed a social interaction tool and a study on how to reason about others.

Guided by these works and the results from preliminary user tests on the matter in hand, we came up with an architecture for our agent. This was implemented on FAiMA’s architecture, taking advantage of part of its core to which we added two more components, our Theory of Mind component and a deliberative component we adapted from another work for our agent.

This agent, capable of detecting deception is represented physically by a social robot-head and used to play Coup against humans with the help of a tabletop running an improved version of a previous implemented Coup game.

To test our approach and answer our hypothesis, we tested our agent against a version of the agent without Theory of Mind while playing Coup against human players (one on one).

From the results we obtained, we get that our hypothesis, "an agent equipped with Theory of Mind shall detect other players' lies more often while playing the game COUP against a human than one without Theory of Mind", was observed, with the agent with Theory of Mind detecting more lies than the one without (48% of lies caught against 1% from the agent without Theory of Mind).

We were glad that, not only did our hypothesis was observed but also that our agent with Theory of Mind detected approximately the same number of lies as the human players, needing less challenges to do so.

Even though we could confirm our hypothesis, the perception of participants over the two agents performance did not differ much. But still our Theory of Mind agent was perceived as a slightly better liar, more competitive and more enjoyable to play against compared with the other agent.

8.1 Future Work

In terms of theory of mind our agent would benefit from a way to make less risky challenges. For example, if the player has a clear advantage in the game, such as more faced down cards ans coins, then there is no point in challenging unless it is almost certain the other player does not have the card.
Another situation in which the agent would benefit from caution is when it loses a card in the first turn but still challenges the other player right after.

On general terms the agent could benefit from expressing emotions, since these could help the agent deceive other players by doing certain facial expressions to react to certain events. For example a simple smile upon receiving a bad hand to make the other player think the contrary.

Another good way to take deception to the next level would be to choosing the utterances in a way that may deceive the other players. This could even complement the expressions, in the given example a "Now this is what I call good cards!" could help sell that idea (or be too obvious on purpose to make the other player doubt it, because it really had a good hand). Faking mistakes would be another way to induce the other player in error, like "I block your Steal with my Duke". The player knows that a "Duke" can not block a steal, so it may believe that EMYS made a mistake and does not have the card to block the "Steal", so it may challenge the block.
Bibliography


Appendix A

Utterances
<table>
<thead>
<tr>
<th>Category:SUBCATEGORY</th>
<th>When is it used</th>
<th>Category:SUBCATEGORY</th>
<th>Where is it used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionStart:FIRSTWIN</td>
<td>At the start of the first session</td>
<td>Counter:ASSASSINATE</td>
<td>When EMYS blocks Assassinate</td>
</tr>
<tr>
<td>SessionStart:SECONDWIN</td>
<td>At the start of the second session</td>
<td>Counter:STEAL</td>
<td>When EMYS blocks Steal</td>
</tr>
<tr>
<td>SessionStart:SECONDLOSTWIN</td>
<td>At the start of the second session</td>
<td>NoChallenge:</td>
<td>When EMYS does not challenge</td>
</tr>
<tr>
<td>SessionStart:LASTWIN</td>
<td>At the start of the last session</td>
<td>NoCounter:</td>
<td>When EMYS does not counteract</td>
</tr>
<tr>
<td>SessionEnd:FIRSTWIN</td>
<td>In the end of the first session if EMYS won 3 or more games</td>
<td>NoCounter:</td>
<td>When EMYS does not counteract after not challenging</td>
</tr>
<tr>
<td>SessionEnd:SECONDWIN</td>
<td>In the end of the second session</td>
<td>ButCounter:STEAL</td>
<td>When EMYS blocks Steal after not challenging it</td>
</tr>
<tr>
<td>SessionStart:SECONDWIN</td>
<td>If EMYS won 3 games</td>
<td>ButCounter:ASSASSINATE</td>
<td>When EMYS blocks Assassinate after not challenging it</td>
</tr>
<tr>
<td>SessionEnd:SECONDCLOSEWIN</td>
<td>In the end of the second session</td>
<td>ChallengeOther:BAD</td>
<td>When the other player does a bad challenge</td>
</tr>
<tr>
<td>SessionEnd:SECONDLOSTWIN</td>
<td>If EMYS won less than 2 games</td>
<td>ChallengeOther:GOOD</td>
<td>When the other player does a good challenge</td>
</tr>
<tr>
<td>SessionEnd:SECONDWIN</td>
<td>In the end of the second session</td>
<td>LoseCard:</td>
<td>When EMYS loses a card</td>
</tr>
<tr>
<td>SessionEnd:LASTWINWIN</td>
<td>If EMYS won 3 or more games and 2 or more sessions</td>
<td>Action:TOOMUCHINCOME</td>
<td>When the other player does too much Income</td>
</tr>
<tr>
<td>SessionEnd:LASTWINLOSTWIN</td>
<td>If EMYS won less than 3 games and 2 or more sessions</td>
<td>Action:TOOMUCHFOREIGNAID</td>
<td>When the other player does too much Foreign Aid</td>
</tr>
<tr>
<td>SessionEnd:LASTLOSTWIN</td>
<td>If EMYS won 3 or more games but less than 2 sessions</td>
<td>Action:TOOMUCHTAX</td>
<td>When the other player does too much Tax</td>
</tr>
<tr>
<td>SessionEnd:LASTLOSTLOSTWIN</td>
<td>If EMYS won less than 3 games and less than 2 sessions</td>
<td>Action:TOOMUCHSTEAL</td>
<td>When the other player does too much challenges</td>
</tr>
<tr>
<td>GameEnd:WIN</td>
<td>In the end of the game (except the last) if EMYS won the game</td>
<td>ChallengeOther:TOOMUCH</td>
<td>When the other player challenges EMYS after not challenging it for a long time</td>
</tr>
<tr>
<td>GameEnd:LOSE</td>
<td>In the end of the game (except the last) if EMYS lost the game</td>
<td>ChallengeOther:TOOFEW</td>
<td>When the other player challenges EMYS after challenging it often</td>
</tr>
<tr>
<td>GameEnd:QUICK</td>
<td>In the end of the game (except the last) if the game was over in 5 or less turns</td>
<td>NoChallengeOther:AFTERTOOMUCH</td>
<td>When the other player does not challenge EMYS after challenging it often</td>
</tr>
<tr>
<td>GameEnd:EASYWIN</td>
<td>In the end of the game (except the last) if EMYS won the game and kept its two cards faced down</td>
<td>ChallengeOther:GOODAGAIN</td>
<td>When the other player has many good challenges in a row</td>
</tr>
<tr>
<td>YourTurn:</td>
<td>When it is the other player's turn</td>
<td>ChallengeOther:BADAGAIN</td>
<td>When the other player has many bad challenges in a row</td>
</tr>
<tr>
<td>Action:INCOME</td>
<td>When EMYS does Income</td>
<td>Remind:FOREIGNAID</td>
<td>When player takes too long choosing whether to block Foreign Aid or not</td>
</tr>
<tr>
<td>Action:FOREIGNAID</td>
<td>When EMYS does Foreign Aid</td>
<td>Remind:TAX</td>
<td>When player takes too long choosing whether to challenge Tax or not</td>
</tr>
<tr>
<td>Action:COUPONE</td>
<td>When EMYS does Coup and the player loses his first card</td>
<td>Remind:ASSASSINATE</td>
<td>When player takes too long choosing whether to challenge/block Assassinate or not</td>
</tr>
<tr>
<td>Action:COUPTWO</td>
<td>When EMYS does Coup and the player loses his second card</td>
<td>Remind:EXCHANGE</td>
<td>When player takes too long choosing whether to challenge Exchange or not</td>
</tr>
<tr>
<td>Action:TAX</td>
<td>When EMYS does Tax</td>
<td>Remind:STEAL</td>
<td>When player takes too long choosing whether to challenge/block Steal or not</td>
</tr>
<tr>
<td>Action:ASSASSINATE</td>
<td>When EMYS does Assassinate</td>
<td>Remind:COUNTERFOREIGNAID</td>
<td>When player takes too long choosing whether to challenge/block Assassinate or not</td>
</tr>
<tr>
<td>Action:EXCHANGE</td>
<td>When EMYS does Exchange</td>
<td>Remind:COUNTERASSASSINATE</td>
<td>When player takes too long choosing whether to challenge Block Assassinate or not</td>
</tr>
<tr>
<td>Action:STEAL</td>
<td>When EMYS does Steal</td>
<td>Remind:COUNTERSTEAL</td>
<td>When player takes too long choosing whether to challenge Block Steal or not</td>
</tr>
<tr>
<td>Challenge:</td>
<td>When EMYS does a Challenge</td>
<td>Remind:NEW</td>
<td>When player takes too long starting a new game</td>
</tr>
<tr>
<td>Counter:FOREIGNAID</td>
<td>When EMYS blocks Foreign Aid</td>
<td>Remind:PLAY</td>
<td>When player takes too long choosing an action</td>
</tr>
</tbody>
</table>

Table A.1: Utterance’s categories and subcategories explained