Lie to Me: Lying Virtual Agents
(extended abstract of the MSc dissertation)

Henrique Reis
Departamento de Engenharia Informática
Instituto Superior Técnico
Advisor: Professor Ana Paiva

Abstract—The objective of this thesis was to endow agents with the ability to model the mind of other agents and to use this information on their reasoning process. We developed an architecture which enables N levels of Theory of Mind. Additionally we wanted to assess if a reasoning ability capable of more then one level would be perceived as more socially intelligent than one with only one level. For this purpose we focused our work on deception behaviour generation. Key concepts from the field of Psychology were reviewed regarding this topic, while some works which tried to implement a similar goal were also mentioned. A case study was developed to compared liars with one level and second level Theory of Mind.

The evaluation itself was composed of preliminary, simulation and a final questionnaire. The results showed to be consistent with our hypothesis that the more reasoning abstraction level of a Theory of Mind, the better agents can performed deceptive tasks.

I. INTRODUCTION

The aim of Artificial Intelligence has been to both understand and build systems with the notion of intelligence. An approach we can take comes from the only natural intelligent “system” we know: ourselves. This fact leads us to one of the main goals of IA: create human-like intelligence. Something that would act by our standards, have the same kind of reactions, display the same emotions, be as fallible as we are. This thesis focuses on this view, with a special concern for social intelligence.

In multi-agent systems, agents interact, cooperate and negotiate with one another. These interactions also need to seem real and rich enough to cast in the user a sense of belief, the “suspension of disbelief” as Joseph Bate first labelled it [1]. While most communications, either verbal or non-verbal, require exchange of information in some way, some of them might be deceitful.

In human-computer interaction (HCI) and MAS, users and agents are both presumed to always say the truth, abiding to the sincerity assertion. However, deception occurs occasionally, both unintentionally and on purpose, exactly as it happens in human communication every day. Therefore deception is one such human-like characteristic that would enrich an interaction and its believability.

Although deception has already been discussed considering its use in AI for a long time, only during the last decade some projects have been developed and implemented it computationally. To enhance the believability of synthetic characters, and endowing them with the capacity of deception, one has to borrow research data from psychology, more specifically from social psychology, highlighting the multi-disciplinary of this specific research work. Specially, a Theory of Mind is needed in this process, so that the agent can attribute a mental state to other agents and be able of reason about it.

A. Problem

In a first stage, the aim purpose of this thesis is to develop a scenario to test lying agents. Therefore it is necessary to give both the motivation and means to lie. Specifically, we want to develop the capacity to lie and behave deceptively in the context of a society of agents, where there will also be other agents that will try to uncover the liars. In this regard, we will focus on a specific problem:

“How can autonomous agents behave deceptively and generate lies relevant to a specific context?”

As we will see, humans resort to a mechanism of more than one abstraction levels to represent others’ mind, which is then used to achieve their deceptive goals. Our work will follow an approach which assumes that modelling what others are thinking is beneficial to perform deceptive tasks. Specifically we believe that an entity A which can represent what an entity B is thinking will be less successful in deception then an entity C which can model what agent B thinks about C. In other others the hypothesis we will try to prove is that:

“The higher the reasoning order an entity is capable to use, the better it can successfully perform deceptive tasks.”

By high-order reasoning we mean that one considers what others are thinking, first level; what others are thinking about others, in a second level, and so on.

We will begin to answer these questions through the analyses and comparison of architectures already developed to achieve a similar goal as this one.

The resulting agent architecture and models will be evaluated with users in the context of a game with a concrete scenario. This evaluation will test the believability of the social interactions, focusing in particular on the deceptive ones, produced by the work of this thesis.
II. BACKGROUND

A. Definitions

To begin the discussion on this topic, we will first adopt the definition by Castelfranchi [2], which considers that deception occurs when a proposition $P$ is induced actively or passively by an agent $A$ to an agent $B$, when agent $A$, the deceiver, believes the proposition is false, or at least does not believe the proposition is true. This definition excludes as being deceptive behaviour cases where the agent $A$ believes proposition $P$ is true, and tries to induce it, while that proposition $P$ false. Taking this into account, lies are speech acts that achieve the same goal of deceit, being only a part of deception. Non-verbal mechanisms can also be employed but we are going to focus our work on deception through verbal mechanisms, or speech acts.

B. Theory of Mind

A Theory of Mind (ToM) is a term coined by Premack and Woodruff [3] which defines the ability to infer the full range of epistemic mental states of others, i.e. beliefs, desires, intentions and knowledge. Having a ToM is being able to reflect about one's own and other's mind content. These complex abstractions we make of other's mind and consequently of our own, is a mechanism that helps us make sense of their behaviour in a specific context and predict their next action.

We do this applying the principle that observing something leads to some epistemic change, which eventually leads to a result, usually is an action. Theory of Mind is the basis of a process called mindreading, and Baron-Cohen describes it as a process we do all the time, effortlessly, automatically, and mostly unconsciously. That is, we are often not even aware we are doing it, until we stop to examine the words and concepts that we are using [4]. Everyday we say that someone wants, thinks, knows, intends, we are ascribing some mental state and trying to infer some coherence out of it. By doing this, we can infer other's mental state, and by observing their interactions we might anticipate what their behaviour will be or even manipulate their actions.

To evaluate if someone has a developed ToM, a typical test to be done is the false-belief task [5]. To pass this test the participant has to be able to attribute a false-belief to someone. An example of this test can be stated as follows:

“There are two boxes A and B on a table, and John puts his ball inside box A and leaves the room. John’s mother then changes the ball from box A to box B to trick John. When John re-enters the room, where will he look for the ball?”

All these actions happen in the presence of the participant. To pass this test the participant has to answer “box A”, which indicates that he can ascribe a false belief to John: that John believes the ball is in box A, while in fact it really is in box B. However, this question only exercises a one level conceptualization of someone else’s mind, John’s mind. A one level Theory of Mind starts to develop in normal children around the age of 3-4 [6] [7] [8], and it is the first step to a fully developed ToM. By the age of 6 years old, ordinary children start to pass tests that exercise their second level ToM [9] [10]. They would be able to answers correctly to the following question, considering the same scenario of John and his mother:

“Where does John’s mother think her son will look for the ball?”

This is a more complex task, where the participant has to be able to conceptualize about embedded mental states. To answer correctly, one has to think about what John’s mother thinks about John. At this stage, children will have the capacity of an adult-like ToM. A second-order ToM gives the capacity to handle most of everyday interactions and social needs.

C. Mindreading

Now that we understand the inner ability which enables humans to be socially intelligent and to perform high-order reasoning about other people’s motivations and beliefs, we must ask ourselves how is this theory used to achieve its purpose. Baron-Cohen proposed one of the most complete theoretical models [4] that consists of four main innate components or mechanisms, which together comprise the human “mindreading” ability (Fig. 1).

Figure 1: Baron Cohen Mindreading Mindreading Mechanisms

The first mechanism is the Eye Direction Detections (EDD), which Baron-Cohen suggests has 3 basic functions: detecting the presence of eye-like stimuli, determining whether eyes are directed to the “self” or other entity, and inferring that if the organism’s eyes are directed at something then it sees that thing. This last function is specially important as it allows the observant to attribute a specific basic mental state like \texttt{Entity-sees-me}. The Intentionality Detector (ID) is responsible for interpret primitive mental states such as goals and desires based on other’s actions. It can represent mental states of the type \texttt{Entity-wants-item}. These mechanisms also allow a dyadic level representation, that is, between the observed entity and the observer or the observed entity and the object it sees. The Shared-Attention Mechanism (SAM) is the third component of this theory and is a high order ability as it is responsible for building
triadic relationships using information from the first two components. These representations depict relations between an entity, the self, and a third object, which can be another entity. Therefore states such as \(\text{Entity-sees-(Entity-sees-Object)}\) can be represented. The last and most important component is the Theory of Mind Mechanism. Its two main functions are to represent the full range of epistemic mental states and to relate them with human behaviour, resulting in a useful and meaningful theory that can be used to reason, anticipate and manipulate the action of others. It makes use of the information provided by SAM component and transforms it into epistemic knowledge using principles such as “seeing leads to knowing”.

III. RELATED WORK

The Related Work Section will cover computational systems that are related to or implement deception. It will begin by analysing a work that takes two approaches to the theory of mind, laying the ground for a general understanding of its requisites and alternative implementations. Then it will analyse three different works that propose an implementation for deception, one in the field of robotics, the other two in virtual environments with synthetic characters. It is important to note that there are not many architectures that focus specifically on this topic, and such there are fewer references to other works.

A. Theory of Mind Implementation Approaches

A Theory of Mind (ToM) explains the ability to define others as intentional agents and to infer their mental states, including their beliefs, goals, intentions and desires. This capacity can then be used to predict behaviour, for social manipulation or to implement a degree of intelligence for virtual training.

Meyer et al. [11] worked on a computational implementation of ToM. Their goal is to endow agents with ToM in the specific context of virtual training. They focused on giving agents the capacity to interact in a believable way with trainees, and to explain their actions and decisions after the experiment is over. The agents model the trainee’s mind and give feedback either by simple action decisions, or by an explanation at the end. This work is interesting because they developed theoretical approaches to computationally implement a ToM. Namely, they focus on two prominent and conceptually very distinct approaches of human theory of mind: the theory-theory (TT) and the simulation-theory (ST). Their difference lies in their elements and how we learn to use them. Both can be implemented by a BDI model [12] because they refer to concepts like goals, intentions, desires, plan and action, elements that can be attributed to a mental state.

According to theory-theory, the mental state we attribute to others is not observable, but is knowable through intuition and insight. Moreover, this ToM theory encompasses a set of principles on how the BDI model concepts interact with each other. These are specific reasoning rules dependent of each inference. The architecture is composed by the typical BDI modules, plus several sets in the belief base: the set of own beliefs and the representations of others’ mental states. Each of this representations incorporate beliefs, goals and reasoning rules, all of which, being in the belief base, are themselves represented as beliefs. Reasoning rules combine beliefs and goals to explain and anticipate behaviour, in other words, to produce plans or actions.

On the other hand, the simulation theory claims that each person simulates being another while trying to reason about their epistemic state. In this account, ToM allows one to mimic the mental state of another person. In practice, what this means is that the agent will use its own deliberative power to reason about other agents. Using this approach the representation of mental states is not strictly defined, thus being able to have the same concepts and elements as the agent’s own mental state, namely beliefs, goals, intentions and plans. The agent can then deliberate about agent’s mental state as if it was his own. The implementation of this architecture can be very modular if the other mental states have the same elements and structure of the agent’s.

In the case study performed by Meyer et al., results showed the only difference lies in the easiness of implementation, as ST models are better in terms of code reusability and modularity. Moreover, the TT model can only deal with BDI models due to their rigid representation of other’s mental state in terms of beliefs, limiting its symbolic representation.

This research work is especially important because it considers and implements two different approaches of Theory of Mind, pointing out its conceptual origins and analysing the implementation undergoing for each.

B. Wagner & Arkin - Deception in Robots

Wagner and Arkin developed some preliminary algorithms to endow an intelligent system with the capacity of deception [13]. Although this work is primarily oriented to robots, it shows an approach for implementing a deceptive framework on an intelligent agent.

In their approach, Game Theory and Interdependence Theory are used to reason about deception and to assess if a deceptive behaviour is needed. This assessment is done considering the social situation an agent is in, which is computationally represented as an outcome matrix. For determining if deception is necessary, they need to characterize a social situation, which is done by locating a situation in the interdependence space. Therefore they try to measure the extent to which an agent’s outcome is influenced by the actions of other agent. Moreover, by setting limits on some dimensions, or defining regions in the interdependence space, situations where deception is warranted can be ascribed.

The first step is to induce a false belief on the other agent. This assumes the deceiver has a model of the mark, which defines the mark’s model. An important point of this work was to examine to what extent modelling the partner can affect the effectiveness of deception itself. The partner model is in practice the theory of mind used to anticipate behaviour.
In this case reasoning about the mark can be done by using the action model and utility functions, which act as reasoning rules.

Both the agent and the mark have the same outcome matrix for a specific situation, the true outcome matrix. The goal of the deceiver is then to perform some action in the environment transmitting a false communication to the mark, that will lead it to behave in a way that would benefit the deceiver. In doing this, the agent is inducing another outcome matrix on the mark, the induced outcome matrix.

Despite some bold assumptions and being a robotics perspective, it shows a way to reason about deception, using Game Theory and Interdependence Theory. It is interesting in the way deception is warranted. The knowledge about our partner, although being a basic model based on features, was also shown to affect the success of a deceit attempt. This specific result is similar to what we want to achieve, that using the information from the Theory of Mind of others, a deceiver can better create a plan to successfully achieve deception.

C. Castelfranchi’s GOLEM

Castelfranchi [2] built a multi-agent world designed to study social attitudes, focusing on cooperation and deception. Deception can occur by conflicts in cooperation scenarios, which is why these cooperative activities are the focus of this work. GOLEM is based on the blocks world of AI planning domain research. Because there are several structures agents can build, goal conflicts will emerge, and each agent will try to achieve their goal, resorting to their own methods or by using the “help” of other agent.

In order to simulate distinct types of cooperation, agents have task delegation and task adoption preferences. These are formalized in GOLEM by a framework of personalities. Agents also have different capabilities, which together with their goals and based on their knowledge of other agents, they use to plan their actions. In this world deception is only instrumental and thus only due to conflicts in the agents’ goals.

In GOLEM, as in the real world, not all information is accessible. Agents only have a limited view of the world and an incomplete and possibly wrong knowledge. This is in fact needed for deception to happen. If every agent have all the information and correct knowledge, there is no room for successful deceptive attempts.

Specific to GOLEM is that deception can be about capabilities, goals or personality, all of which come ultimately form conflicts in achieving goals.

This work is worth mentioning because it has several similarities to the goal work of this thesis. By creating a simple multi-agent world, basic concepts of cooperation and deception could be analysed. Also, developing a personality framework to model agent tendencies is a similar approach to what FAtiMA architecture proposes in Section V-A. This traits influence the agent’s actions through a reasoning process that takes in account beliefs, personalities and capabilities, which is what the agent’s in GOLEM can lie about. However agent in GOLEM can only lie in this limited scope, they cannot lie about their requests for example. This would require a second order reasoning that would endow the agent with the capacity of reasoning about the reasoning of other agents.

D. Mouth of Truth

De Rosis and Carofiglio take another approach for implementing deception [14]. They try to focus on the communicative perspective of a deceptive action, while trying to test their approach on a simplified version of the turing imitation game. In their specific scenario the sender tries to convince the receiver that some fact X is not true, where the sender can lie or use other deceptive strategies. This case study is very oriented to communication. They tried to develop a way to undermine the receiver’s belief, increasing its doubt. Mouth of Truth implements reasoning models as belief networks [15, 16], and beliefs as their nodes. Uncertainty is represented as usual, by probabilities associated with other nodes. In this way, beliefs are connected in a network, where a fact implication logic can be applied, enabling deception by telling a lie about a belief that is not what the sender wants to manipulate, but a belief that is connect to that one. For example, uncertainty can be given to the belief “it rained” if the sender says “the floor outside is dry”. However, the agent also needs to have some mental image of the receiver to be able to deduce this. Therefore, it is assumed the mental state of the agent is composed of a set of its own beliefs and reasoning rules, and the mental state of the other agent, with the same elements. The receiver’s beliefs can then influence the decision making process of the sender. This constitutes the theory of mind of Mouth of Truth, which takes a theoretic approach, as mentioned in Section III-A.

The planning is very similar to the one used by Wagner and Arkin in their deceptive robot (Section III-B). The outcome of this planning process is the best action the sender can perform to manipulate the receiver’s belief, which is then validated considering a number of parameters like impact on the desired belief, plausibility and credibility of the deception behaviour, safety about not being discovered, and computational cost.

The Mouth of Truth approach is a different perspective on how to implement a deceptive behaviour mechanism, because it focus on the manipulation of uncertainty of beliefs. These beliefs compose the agents’ mental state together with a set of reasoning rules. Because the mental image of the receiver also has the same structure, hypotheses have to be made to define these rules for it. This problem is solved using the restrict domain of the experiment scenario: each agent only has the other as a source of information, thus the sender models the receiver’s reasoning rules using its own set of rules. This leads to a limited simulative perspective of ToM in practice, while conceptually it uses a theoretic approach, as described in Section III-A. Also to note is that this specific approach could not be implemented using a BDI model because there are no concepts like desires
and intentions. However, it is the conceptual ground that is interesting to analyze and have in mind, that deception is a communicative act that exploits the uncertainty of beliefs.

IV. A MINDREADING ARCHITECTURE

In this section we will describe the model we have designed to overcome the problem our work tries to solve: “How can autonomous agents behave deceptively and generate lies relevant to a specific context?”

We believe that modelling other minds is required to achieve complex deceptive behaviour, hence following our hypothesis that the higher the reasoning abstraction level regarding other agents thoughts the better the results can be achieved in deceptive tasks.

In order to build agents that can cope and behave intelligently in a social environment, we propose a model which endows agents with Theory of Mind mechanisms (fig. 2). Our view is based on the Mindreading model by Baron Cohen, explained in Section II-C. This architecture follows the BDI model approach of Simulation-Theory discussed in Section III-A.

![Figure 2: Proposed Conceptual Model for a Theory of Mind](image)

An agent is defined by an Agent Model. The Knowledge Base (KB) is the structure where all the propositions the agent knows to be true are present. It represents its beliefs and its knowledge about events that have occurred. This information is the foundation for the agent’s behaviour, as he acts upon the knowledge he has of the world. The Theory of Mind Mechanism (ToMM) component is responsible for modelling and managing other agent’s mind representations. The agent will take this information into account when he wants to manipulate someone else. Finally, the Deliberative Means-Ends Reasoning component defines the planning capabilities of the agent. It is responsible for using the represented knowledge, both in the KB and ToMM to place there.

The implicit purpose of our model is to extend a first level Theory of Mind approach, mitigating its flaws. A one level Theory of Mind agent can only model what others are thinking about and not further than that. This is specially notorious in deception scenarios because it gives the capacity to counter a Theory of Mind one level lower. We have seen that deception requires the need to explicitly want to change the mental state of another.

B. EED and SAM Mechanisms

Our model preserves the Mindreading model of Baron-Cohen only to some extent. The EED mechanism is still present but was generalized to represent all types of sensorial perceptions, although its purpose still remains mainly on what agents are seeing. We assumed that if an agent is within a certain radius or in the same place, it perceives all the present objects and entities, as well as all events that take place there.

The output of EED is a dyadic representation of the type ⟨Ag⟩:perceives:(Proposition), which reflects that the agent ⟨Ag⟩ has perceived a proposition ⟨Proposition⟩, be it an object, event or another agent. Its output is used to update the agent own KB, first level ToM and as input to the SAM mechanism.

The SAM mechanism is fed by the output of the EED component and computes its output in order to determine which minds perceive what, creating triadic relations of the type ⟨Ag1⟩:perceives:(Ag2): perceives:(Proposition).
The dyadic representations directly update the self KB and the first level models. Let us consider that \( \langle Ag1 \rangle \) perceives \( \langle Ag2 \rangle \):perceives:(Proposition). Being a dyadic relationship, we would select the mind of \( \langle Ag2 \rangle \) in the first level of the ToMM component of \( \langle Ag1 \rangle \), and then update its KB. If \( \langle Ag1 \rangle \) has however perceived a triadic relationship like \( \langle Ag2 \rangle \):perceives:\( \langle Ag1 \rangle \):perceives:(Proposition), we would need to update a model in the second level ToM. \( \langle Ag2 \rangle \) would be used to select the model at the first level, and using its ToMM component we would select \( \langle Ag1 \rangle \), proceeding to update its KB. This way \( \langle Ag1 \rangle \) would be able to represent the knowledge that he believes \( \langle Ag2 \rangle \) believes we knows about \( \langle \text{Proposition} \rangle \).

C. Creating and Updating Models

It is important to address the topic of how are models created. The EED is used to perceive other entities. Whenever the agent sees a new entity in the environment it initializes a new model to represent it in its own mind, ToMM component. The memory will begin with no information and will be updated throughout the simulation, as the agent perceives the other agent receiving perceptions itself. When the agent receives an event it will update its beliefs and memory. That same perception is used to update the models of others in the same way, simulating the changes in their internal states.

D. Deliberation and Means-Ends Reasoning

The second purpose of a Theory of Mind is to be able to make use of the knowledge it represents. Modelling explicit goals that involve changing the mental state of others cannot be achieved otherwise. The Deliberative and Means-End Reasoning Component (Deliberation Component in short) makes use of conditions to test the state of the world, either the property of an object or entity, or by testing events that have occurred. A deliberative process without Theory of Mind would only need conditions that are confined to the context of the agent’s own mind, its own KB. Our work proposes that each agent also stores models of others, and so we need to be able to specify that on conditions as well. Extending them to model dyadic and triadic relation achieves the intended purpose: \( \langle Ag \rangle \):knows:(Proposition) and \( \langle Ag1 \rangle \):knows:(Proposition) and \( \langle Ag2 \rangle \):knows:(Proposition), which we have used naturally before. Any time the Deliberative component finds one of these conditions, it will first verify which ToM it has to select to test the condition against. If a condition does not specify a relation it will be tested against the agent’s own model, in other words, its own KB.

Let us imagine John wants to trick Mary into believing that a candy is spoiled. The desired state of the world is that Mary believes Candy(isEdible) to be false, while John believes Candy(isEdible) to be true. These conditions, called success conditions, need to be verified for John to achieve the goal. The first condition is true if according to the model John has of Mary the candy is not edible, in other words Candy(isEdible) is false, while the second condition is true if it holds in John’s own KB.

It is also important to note that the planning mechanism needs to support this kind of condition and effects that refer to a certain model of Theory of Mind. First, these goals have to be selected and then be computed by the planning mechanism to create a plan. Plans are made of a set of actions, which upon done successfully achieves the selected goal. In order to create plans we need a planner. Actions are operators, defined by tuples OP\( \langle Ag, N, P, E \rangle \), where Ag is the agent who executes the action, N the name of the action, P the set of condition that need to be verified before this goals can be done, its pre-conditions, and E a set of conditions that hold after the goals is achieved, its effects. We need to allow the goal’s pre-conditions and effects to represent dyadic and triadic relations, for example SELF:knows:Candy(isEdible) and SELF:knows:Mary:knows:¬Candy(isEdible).

Typically there are two types of effect we need to be able to represent: global effect and local effects. The first type of effects represent those that are known to every agent, and can be represent as \(*:(\text{mental-state})::(\text{proposition})\). For example, \(*:\text{knows}:Candy(isObject)\) is such an effect. The second type of effects represent those that are only known to a specific entity and have a localized effect. Dyadic and triadic effects can then be defined like \(\langle Ag \rangle:(\text{mental-state})::(\text{proposition})\) and \(\langle Ag1 \rangle:(\text{mental-state})::(\text{proposition})\). An example of each of this cases can be Jonh:knows:Candy(isEdible) and Jonh:knows:Merry:knows:¬Candy(isEdible).

V. IMPLEMENTATION

Having explained the model for endowing agents with representative abilities to model the minds of others, we will now explain how we implemented the aforementioned mechanism. The frameworks and architectures with which the model was integrated will be explained, with special focus on how the update mechanisms were performed. We finally proceed to a full description of the case study we developed to test the validity of our model and its ability to generate deceptive behaviour.

A. FA\(\text{iM}A\) Modular Architecture

F\(\text{A}\)\(\text{iM}A\) [17] [18] is an autonomous agent architecture implemented in Java\(^1\) with planning capabilities which we used to integrate our model. Its purpose is to build virtual characters which behave and reason in a way that it is influenced by their emotional state and personality. Their behaviour is meant to be believable and create empathy reactions to the users. Because each agent can potentially have a different personality from one another, their reasoning will differ according to the situation. In such cases an emergent effect can occur, where actions and reactions unroll creating a flow of interactions that was not explicit intended. The personality of an agent is defined by the following set of traits: (1) emotional threshold and decay rates, which affect how emotions evolve; (2) a set of goals, that the agent can possibly pursue; (3) a set of emotional rules, which defines the emotional response to an event; (4) a

\(^1\)www.java.com
set of action tendencies, in other words, reactive actions to an event. Therefore, each agent will subjectively evaluate a situation based on its personality and behave differently according to its goals.

These approaches have set the guidelines for the architecture’s development, which is now composed by two main complementary parts: FAtiMA Core and FAtiMA Modular Components.

B. FAtiMA Core

FAtiMA Core, as the name states, is the backbone that provides the basic algorithms that generally define the core aspects of its functionality.

![FAtiMA Core Architecture](image)

The affective state stores the emotions of the agent and is responsible for their decay, which can be modified by the appraisal process. In the memory is outlined all the knowledge and event occurrences that the agent as perceived. The action selection mechanism ultimately defines the agent behaviour, through its actions. It is important to mention that FAtiMA Core does not commit itself with particular implementations of each of these mechanisms, aside from the memory component.

The agent is in an endless loop processing events and updating its internal state. FAtiMA Core is only responsible for notifying the components about new events, and making use of their output to trigger the agent’s behaviour through the execution of actions. The agent can either receive external of internal events to begin this process. An external event is a change in the environment, while an internal change corresponds to events in the agent’s internal architecture, for example, succeeding or failing a goal. Upon receiving an event, the memory is updated and the agent is now aware that event happened. At this stage, inference operators specifically defined are triggered to test the new state of the memory and possibly infer new knowledge. Each component is then updated with the received perception. Finally, an action is chosen and perform by the agent.

C. FAtiMA Modular Components

Agents that are only defined by FAtiMA Core will not do anything because modular components need to be integrated to the system. They must implement the relevant interfaces to add specific functionalities. Each component can implement one or a subset of these mechanisms, as it is not required to implement all of them. Upon initialization of FAtiMA Core, each component is added depending on that specific agent instantiation definition.

We will proceed to discuss the most relevant components used in our work.

1) Theory of Mind 1st Level: FAtiMA Modular already has a component that implements a first level Theory of Mind. As mentioned in Section II-B a ToM has two main functions. The first one is to represent models of other agents, including their beliefs, motivations, emotions and past experiences. This is easily achieved by replicating the agent’s own model structure, taking the simulation approach as described in Section III-A. The agent will therefore model each of the other agents in the environment as a replication of himself. Thus, both the agent’s own defining class, AgentCore, and the class which defines other’s models, ModelOfOther, implement the interface IAgentModel so that they can be treated equally during the agents deliberative process.

Although the basic structures are the same, as we have explained in Chapter IV, they have some small simplifications mainly regarding components. Because this is only a first level Theory of Mind, models of others do not have a ToM component. Another big simplification made, as mentioned in Section IV-A, was that model of others do not represent goals, intentions and reasoning components.

Models are updated in the same way we explained in Chapter IV. Due to simplification reasons, agents perceive events that happen close to them, implementing a simple approach to the EDD component mentioned in IV.

To use the stored knowledge in ToM model the agent need to have explicit goals regarding minds of other. Usually these would be tested against the agent’s own model and knowledge base. Since we also want to use information stored in internal states of model of others, the first step was to enable the representation of condition in terms of model of others, so they can be used like normal conditions as the agent itself. This is represented in XML by adding the tag “ToM” to conditions.

This can either be verified if the proposition is false in the memory of Mary’s agent model, or if the proposition is simply not in the KB. A property like that can be part of a goal’s pre-conditions or success conditions, thereby making use of other’s mental state in the reasoning process of the agent. Whenever the Deliberative Component finds a condition it will test it against the relevant agent model. If no ToM tag is specified, the component will test it against the agent own structures, otherwise it will ask the ToM Component for the correct model and test it then.

There is also two type of effects that have to be accounted
in planning operators, Global Effects and Local Effects. Global Effects are those that are perceived by all the agents in the scene. This type of effect does not have a ToM tag specified. Local Effects are effects that are only perceived and acknowledge by a particular agent specified in the ToM tag.

D. Implementing a Mindreading Agent

Although the Theory of Mind component already exists implemented in FatiME, it does not completely fulfils the purpose it could, as we discussed in Chapter IV. The capacity of Theory of Mind Component is limited by having only one level.

1) Theory of Mind - Second to Nth level: The basic rationale behind our work is to create nested levels of models which represent the other agents in the environment. Eventually N levels can be represented, but as we saw in Chapter IV its complexity grows exponentially with the number of levels. If we remember that each one of these models are treated as the agent's own and updated with the events in the environment, the overall performance of the system would be severely jeopardized. Therefore, we will focus on second level Theory of Mind, taking into account that the abstraction level could be enhanced in exchange for the agent's responsiveness speed. Third and fourth levels were also tested. However, these levels are much more complex to both understand and test, and thus we focused on the second level.

2) Creating Model Of Others: When the simulation begins, the agent receives a list l with m agent names, representing all other entities in the scene. In the first level ToM version we would only need to create a model for each of those agents and store them in the ToM Component. Now we want to create models to represent N levels of Theory of Mind. The specific number of levels, a variable called maxToMLevel, is received as an argument of the ToM Component upon its initialization. The rationale is to recursively initialize all models, creating a tree-like hierarchical global structure. Every model would be initialized upon creation, along with all its components. The decision of what components to instantiate is the result of the following process: for each component used by the parent node, if it implements the interface IModelOfOtherComponent then it is recreated in the new model.

However, it is important to note that when a model is being initialized in a level equal to maxToMLevel, ToM Components will no longer be created.

After the instantiation process ends, the agent will trigger an innate objective to perform the "look-at" on all the entities he knows of, in other words, the agents corresponding to the models in ToM’s first level. Thus, the following perceptions will be related to properties of others.

However, we did not want to limit this initialization stage to a one-shot process. If we imagine that agents A, B and C are in a room, a similar process would have to trigger if an entity D would enter the room in the midst of the simulation. That entity could be the User for example. In such a case, agent A would receive an “entity-added” perception telling him there is a new entity, and a similar process would ensue: agent A would “look-at” the User, receive its properties through perceptions, update its knowledge base and the ToM models.

3) Updating Model of Others: The agent will start by looking at all the entities in the scenario and will perceive their properties, updating its memory accordingly. Let us take the following example: John and Mary want to eat the candy that is on the floor. As mentioned, we will use a second level of Theory of Mind.

The original mechanism only updates up to the first level of ToM. How can we propagate to further levels? The answer is to further turn the mechanism recursive. Therefore, every module should be updated independently and individually with each perception the agent receives, and we will do just that. In the first level ToM version, the component would just take every property of target that was seen and duplicate it in the model which corresponds to the subject that perform that action “look-at”. Because we have assumed the theory that “seeing leads to knowing”, we can infer that if someone saw something, then it knows about it. We further infer that people would know the same as we do. To expand this notion, we perform a recursive update on the relevant models.

4) Inferences: Inferences also perform an import rule in the updating process of agent models. They enable the implementation for the SAM Component mentioned in IV-B. Inferences are defined in the agent authoring files as a special type of actions, which will not be executed but rather triggered when new information is added to the agent’s memory. As we have mentioned in the beginning of this Section we have adopted a simulative approach, whereby each model has an independent reasoning process. This also applies to the inference process. We have changed this specific mechanism to be done independently by each model as well. Regarding the ToM component update, it will proceed in a similar fashion, updating every model sequentially. Upon being updated, if any new knowledge was added to the model’s memory since the last update, the inference process will trigger and all operators will be verified.

VI. Case Study

This case study was implemented using the ION Framework [19] to manage the world simulation engine, and the Never Winter Night 2\(^3\) game engine for the graphical interface. We aimed for a testing scenario were bluffing, deception and manipulation were core aspects of the game.

The Werewolf/Mafia game\(^4\) was chosen to make a scenario where characters have incomplete information and vision of the world. The core game-play evolves around the

\(^{2}\)http://www.obsidianent.com/
\(^{3}\)http://www.atari.com/
\(^{4}\)http://en.wikipedia.org/wiki/Mafia_(party_game)
notion of trying to either hide or discover hidden information through interactions with other players.

We adopt a very simple version of the Werewolf game. There are five players in the game, called villagers, who are divided in two groups: the Werewolves and the Victims. Our version focused on a one Werewolf scenario, thereby existing four remaining Victims.

Villagers have limited information about the environment, since they don’t know who is the Werewolf. Therefore, their goal is to discover who is the Werewolf among all villagers, in other words, the one who is lying. On the other hand, the Werewolf knows his role and subsequently who are the victims. Having all the hidden information about the villagers’ roles, its objective is to remain hidden until he is not outnumbered by Victims, thus trying to eliminate all Victims while concealing its true identity.

The game progresses in a turn based manner. Each round every villager will perform the Accuse action in order, targeting the villager they think is the Werewolf. As one might guess, the Werewolf itself will try to deceive by accusing a Victim.

At the end of each turn the villager voted by the player’s majority will leave the game, being “lynched in the public square”. Our version of the game has children playing. When one of them loses they just leave the area, not actually dying. Upon being excluded from the game the villager informs other agents about their true identity. This information can be used to infer new information, usually regarding past accusations.

The Werewolf will win if it reaches the last turn alive, at which point there will only stand two villagers: himself and another Victim. At this stage he could reveal his true nature without retaliation and win the game. Victims win if they manage to discover who the Werewolf is before the last turn.

VII. Evaluation

The final evaluation we performed aimed to test all the assumptions and theories we have discussed throughout this document. In particular, we wish to test the validity of our hypothesis, which states that:

“The higher the reasoning order an entity is capable to use, the better it can successfully perform deceptive tasks.”

Although our evaluation is based on a game, due to time and resources restraints we have chosen to not make an interactive experience test between the user and the agents in the game. Such a test would require more time to be performed, hence gathering fewer potential results than an interactive one.

The evaluation was performed through online questionnaires. Individuals received invitations on Facebook5 and e-mails directing them to our evaluation. The same questionnaire was applied to two different video demonstrations, each with our test conditions: ToM1 with a liar using a first level Theory of Mind; ToM2 using a second level Theory of Mind. Participants were randomly directed to one of these versions. Each participant only applied for one version and not both. A total of 60 participants (56.67% males and 42.6% females) took part in our evaluation.

A. Results

Our analysis shows that participants did in fact perceive ToM2 condition as being more interesting and would play such a variation of the game with significance results between conditions. We saw that our conditions introduced a change in the game’s overall difficulty to each role. Participants thought ToM2 variable had increase the difficulty to Victims while eased the task of the liar. We can conclude the liar did a more proper job in the second level Theory of Mind version, which applies of the test perceived as of increase of difficulty for the Victim role.

The player’s predictability was also significantly influenced by ToM2 variable ($p < 0.001, |r| < 0.5$). Participants perceived the control variable, ToM1, has less predictable than ToM2. We believe this is influenced by the liar actions. While trying to deceive, the liar should not give himself away. Fostering predictable behaviour through manipulation the liar in the ToM2 variable achieved its goal with more benefit, winning in fact.

The easiness in deceiving is highly influence depending on which variable we use ($p < 0.001, r = -0.478$). Players were deceived more easily using ToM2 variable, which is supported by a high significance and a close to large effect size. This means the difference between results comes greatly from our testing variable and not chance. We must remember that our variables account for a difference the liar’s behaviour and not that of the victims, by which we can conclude that ToM2 condition did in fact introduced a factor which changed the results, i.e. a better deceiving agent.

Measures about “how well did the liar played” (Q12) and its intelligence (Q13) resulted in statistically significant differences between conditions ($p < 0.001$) supported by a large effect size($|r| = 0.5$), meaning our test variable, ToM2, had a large impact on the control variable, ToM1. We concluded that the liar from variable ToM2 is more intelligent than the one from the control variable. Also important to note due to its high statistical significance and close to large impact ($p < 0.001, r = -0.467$) is the fact that the liar in ToM2 variable was affected by other players’ action.

Results also showed that the liar in condition ToM2 managed to deceive to a better extent then test condition ToM1 ($p < 0.001, r = -0.524$).

Our aim was to test if the higher the reasoning order one can use, the higher the proficiency in deceptive tasks. We can conclude that our results not only support our hypothesis, but also supporting that higher order reasoning is a clear sign of social intelligence.

5www.facebook.com
VIII. CONCLUSIONS

This dissertation tries to address the following problem regarding social intelligence in IA:

“How can autonomous agents behave deceptively and generate lies relevant to a specific context?”

This is a kind of behaviour we only encounter in social environments, hence being important to build in autonomous agents since we are aiming for complex social interactions.

We argued that an important step to achieve that goal is through abstraction reasoning, namely using a mechanism humans use, called Theory of Mind (ToM). Several research works have implemented this ability in agents but have only addressed a simplification of the problem, considering only a one level Theory of Mind. In this way, an agent cannot reason about what others are thinking about him for example.

We propose that the problem above can be tackled with the following hypothesis:

“The higher the reasoning order an entity is capable to use, the better it can successfully perform deceptive tasks.”

We developed a conceptual model to achieve N levels of Theory of Mind through a simulation approach, where the agent reasons about what others believe and do as if he was in their shoes. This model is based on a complete theory about how humans ascribe mental states to other people, called a mindreading ability.

The evaluation to our work was done by online questionnaires done by 60 participants, comparing a scenario with a liar using second level ToM and another scenario with the liar using third level ToM. The results showed that participants clearly perceived testing condition ToM2 as being best at deceiving other agents. An interesting result shows a clear relation between what participants perceived as being best at deceiving other agents. Testing condition ToM2 was both considered more intelligent and better at deceiving other agents than ToM1. This result confirms our prediction that deception is a necessary trait in social intelligent agent.

All in all our work showed good results and supported our hypothesis that a higher level of Theory of Mind results in better performance of deceptive behaviour. Due to the modularity of our architecture, it can be configured to implement N levels of ToM, thus being able to be used to configure complex scenarios and be used in other research works related to ToM.

REFERENCES