Strategically misleading the user: building a Deceptive Virtual Suspect

Diogo Rato
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
diogo.rato@tecnico.ulisboa.pt

ABSTRACT

Humans lie every day, from the least harmful lies to the most impactful ones. It is part of the strategies we use in our daily interactions to deal with conflicting situations like negotiation or sharing compromising information. Therefore, in an attempt to design virtual agents endowed with advanced decision-making abilities, researchers not only focused their effort in designing cooperative and truthful agents but also deceptive and lying ones. In this paper we propose a model capable of engaging an agent in an uncooperative misleading dialogue with a user. This model gives to an agent the ability to reason about its knowledge and then autonomously adjust the story it tells depending on what its interlocutors might know and on how sensitive it considers the conversation topic to be. Such a model allows a story's author to focus on the main narrative, letting the model handle the generation of alternatives. We implemented the model in an agent called the Deceptive Virtual Suspect, used in an Interrogation Game. The game portrays an agent acting as a criminal suspect trying to mislead the police by concealing information about its past events. Using the Interrogation Game, a preliminary experiment was conducted to investigate the conditions in which the agent produces more lies. The results showed that participants who used interrogation skills similar to the one the police use, that is, asking open questions first and closed questions after, made the agent adapt its story more often, leading the interrogators (the users) to identify more easily the deceptive behaviour of the agent.

CCS Concepts

•**Computing methodologies** → **Distributed artificial intelligence;** Intelligent agents;

Keywords

Adaptive Storytelling, Human-Agent Interaction, Deceptive Communication, Autonomous Agents

1. INTRODUCTION

As humans, we use deception daily as a tool to cope with conflicting situations in a variety of contexts. From small lies, with less impact in our life, such as lying about leaving a window opened in a rainy day [7], to more serious offences, like having an affair or participating in a robbery [6].

As a result of being so frequently used in conversations, deceptions could be incorporated in virtual agents to make them more believable, mimicking how human choose to tell the truth or to lie. An autonomous agent's model with such deceptive capabilities could be applied in different situations, such as in tutoring scenarios aimed at teaching trainees techniques to detect deception, in interactive storytelling in order to create a richer plots, or to allow NPCs in video-games to mislead the player and create a challenging and more rewarding experiences.

Despite its extensive usage in our dialogues, lying is not a straightforward task. It requires a higher effort than telling the truth [20][21], since, in order to inhibit cues that could expose it, the liar needs to mentally keep track of the lies in parallel with the real events [26]. Additionally, the deceiver must also be able to share information consistent with his interlocutor's knowledge[9].

Moreover, little is known about the cognition of deception [9]. A model to describe the cognitive process used to produce lies was proposed by Walczyk et al. [24] and later expanded and redefined for deception involving high stakes [23]. The model, called Activation-Decision-Construction-Action Theory (ADCAT) proposes the following steps: first the truthful memories are activated, then a decision is made whether there is a need to change the information shared. Finally, a deception is constructed based on this decision and the deceptive statement is shared.

Regarding the scope of deception, people are able to lie about almost everything. Thus, trying to mimic such broad capability in a virtual agent may seem unattainable at first. However, researchers have started to design and implement deceptive virtual agents to be used in closed and well-defined environments. Recently, some researchers have been investigating the development of virtual agents endowed with deceptive techniques applied in negotiations. In environments where resources need to be shared, conflicting goals appear and autonomous agents are forced to negotiate to achieve their goals, either by using cooperative or competitive strategies. In such a particular setting, autonomous agents capable of deceiving others about their capabilities, goals or personality can hide their true intentions to serve their self-interest [4]. However, while deceiving, the agent needs to maintain a sense of fairness within the other to help him achieve its goals [10]. Filtering the information available to others and sharing useless information can increase

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

the agent's chance of misleading its opponents. This highlights the importance of balancing the amount of irrelevant and relevant information given to an interlocutor in order to ensure that he will not start to ignore it [11]. These examples illustrate how autonomous agents can benefit from the use of deception and how balancing the amount of shared information is important when creating deception. However, these works focused mainly on negotiations.

In our work, we want to explore the deceptive mechanisms involved in a task of sharing past actions. In other words, we want to explore the process of telling a story and how to automatically build alternatives in order to hide true information. To deceive others about its previous actions, our virtual agent needs to represent a narrative in its memory and should have the inference capabilities needed to autonomously adjust it as it goes along.

Tulving proposed that memory could be divided into two groups [18]: Episodic Memory that stores perceptual information about dated events and temporal-spatial relations among them and Semantic Memory that organizes knowledge about words, concepts and rules [19].

Episodic memory has been used by virtual agents capable of recalling their past experiences and beliefs [8], but also by artificial companions to model shared memories gathered from a dialogue with users [3]. In their representation, an episode (i.e. a fragment of the agent's memory) contains information about its time, location, the people and other entities involved. When these episodes are grouped sequentially, they correspond to the autobiographic memory as defined by Conway [5]. While allowing a rich description of the episodes and their organisation in a timeline, these models propose a static representation of the memory. For a virtual agent to adjust its story during an interaction, this one needs to be dynamic and to adapt when required.

Therefore, our objective share similar challenges found within the field of interactive storytelling and our agent's model is inspired by the story adjustment mechanisms found in this literature. In the Narrative Mediation [15], the authors proposed a description for the story as a linear progression of events where the user actions are anticipated. In Game AI as Storytelling [16], the authors introduced *author goals*, a technique used to manipulate the plot of a game in order to meet the game designer's intention. These approaches rely on the story's author to explicitly handle the user's actions and to create alternative paths for each one of them. Inspired by these previous works, we plan to integrate relevant aspects of the referred functionalities to give the story's author the power to implicitly guide the deception created.

Few works have explored deceptive strategies for an agent in the context of a police interviews. One work particularly interesting for us is the Virtual Suspect's Response Model in which the authors studied different aspects of the question-answering interaction that occurs during police interview [1]. They presented a model describing the question and answer structure along with cognitive components that decide if the virtual agent shares details about its actions based on its personality and the interpersonal relation it has with the interviewer [2]. Building from a similar architecture for the question and answer structures, our agent differs as it will autonomously adjust and share details of a story with the user based on the questions asked and the state of its memory, disregarding the affective and emotional aspect of the interaction for now.

Overall, despite sharing some goals and challenges of previous research, our work focuses on designing a virtual agent's model that is able to autonomously create, adjust and share false information about its past actions. One of the key element of our approach is that the agent is manipulating in its mind a representation of its interlocutor's knowledge in order to try to produce alternatives that are consistent with what the interlocutor already know. One of our challenges is to allow the agent to maintain parallel stories, the original one and the adapted ones, and to keep track of what information have been shared with each interlocutor. Additionally, along with enhancing the social interaction skills of our agents, our approach intends to reduce the authoring process associated with the creation of scripted alternatives to deal with every possible user's action, allowing the story's author to focus on the development of the main narrative.

With the expertise gathered from different fields and the objectives set, we defined the following problem: *Can we design and implement a virtual agent's model capable of creating and maintaining a false story during a conversation?*

Our objective is given an agent A, a interlocutor I and a real story S_r, being a story a sequence of related events, A will be able to share a story S_f that has events not contained in S_r and is coherent with a story S_i, a subset of S_t, that represents the story I knows. During the interaction, S_i will have more events from S_r based on the question I asked, and A will create and adjust false events in S_f that are coherent with S_i.

We developed an Interaction Game where a virtual agent equipped with our model acts as a suspect in a police investigation and where the user has to question the suspect to discover details about the agent's story. To demonstrate the model's functionality, we conducted a study using our implementation where we collected several question sequences and evaluated them based on police protocol's rules, allowing us to compare the Deceptive Virtual Suspect story adjustments based on the quality of the questions asked.

To address the problem defined we structured the paper as follows. First, we present the structure of the story representation that will be manipulated by our model. Then, we describe in detail how our model, starting from a question, produces an answer that can contains true or false information and how its architecture endows a virtual agent with the capability to manipulate its memory structure. After this, we present our implementation, the Interrogation Game, that allows a player to interrogate a Deceptive Virtual Suspect equipped with our model. Then, we present the results of an evaluation we conducted using the Interrogation Game. Finally, we conclude our work by discussing the model's limits and its perspectives.

2. STORY REPRESENTATION

The model's main element is its story. Since its content will change during the interaction several times, its representation needs to support the creation of alternative stories.

Our story structure is based on the episodic memory theory [18]. Each memory fragment describes a time period of the agent's life, which are called *events*, and contains multiple *entities*. An *entity* represents a concept within the model's domain that can be referenced by different events.

It is defined by the following tuple:

$$\langle id, value, type \rangle$$

The field *id* uniquely identifies the *entity* in the story, the second field represents its value and *type* is used to classify similar entities. For instance, to represent a "Revolver" of type "Gun", the entity that represents this concept is defined by:

$$\langle 1, Revolver, Gun \rangle$$

An *event* can associate multiple entities in six different fields: *Time*, *Location*, *Agent*, *Theme*, *Reason* and *Manner*. The fields *Time* and *Location* are mandatory, the others are optional and may contain multiple entities. Additionally, an *event* is characterized by an *id* as well and three additional fields that will be used in the deceptive mechanism: *real*, *sensitive* and *action*. The field *real* will take as values *real* or *false* depending if the event really occurred or is a creation of the agent's lies. The *sensitive* field is used by the story's author when creating the agent's memory to indicate with a percentage which event will be considered compromising. The last field represents the *action* associated with the event. For example, a real and very sensitive event in which "John Doe", *entity 2*, bought a "Revolver", *entity 1*, at "Guns and Knife Shop", *entity 4*, on October the 4th, *entity 3*, would be represented as following:

$$\langle 5, real, 100, \text{"Buy"}, 3, 4, \{2\}, \{1\}, \{\}, \{\} \rangle$$

A *story* is a collection of *events* and, in order to share the events between each alternative *story*, a same event can be referenced by multiple stories. Furthermore, this referencing mechanism is dynamic as a story can remove and connect events when needed, giving the story representation the adaptability required to change its content. Additionally, to create a wider range of possibilities during the lying phase, the model's domain may have entities that do not appear in any event and false events that are not referenced by any story. The following section elaborates on the mechanisms used to modify the story representation in order to mislead an interlocutor questioning a virtual agent that implements our model.

3. DECEPTIVE AGENT MECHANISM

An agent equipped with our model interacts with an interlocutor using a turn based question-answering system: the user asks a question, the agent autonomously adapts its story, then gives an answer to mislead the user, and the cycle goes on.

A question is described as a *query* and has the following structure:

$$\langle question\ type, question\ focus, question\ conditions \rangle$$

The first field is the *question type*, which can take as values either *validation question* or *information-gathering question*. These types are based on the two major classes for question classification, *yes-no* and *wh*-questions, proposed by *Quirk et al.* [14] and later studied by [25] and [17]. The second field, *question focus*, corresponds to the elements of the story the interlocutor, the one asking the question, wants to know (similar to interrogative adverbs). The third field, *question conditions*, is a list of conditions that will be used to query the system and to match the results the user is looking for. Each condition is defined by the tuple $\langle field, operator, value \rangle$.

Using the *query* representation, the question "When was Peter at Home?" would be represented as a *information-gathering question* that focuses on retrieving the *Time* field of the events that satisfy two conditions, one defining the events *location*, "Home", and another for its *agent*, "Peter".

When a virtual agent that implements our model receives a question, it processes its content to adapt the story shared with this interlocutor based on its knowledge and how sensitive are the events satisfied by the *question conditions*.

The virtual agent's model proposed implements this mechanism using a vertical layered architecture with its core being the **Query Engine** (described in Section 3.5). In conjunction with this engine, the three layers of the architecture support the real-time modification of the story during the interaction. These layers are the following: the **Theory of Mind Layer**, the **Strategy Selection Layer**, and the **Story Adjustment Layer**. Each layer has its own functionality, but they all use a common **Knowledge Base** where the stories and information about each user are stored. Figure 1 shows the different layers, the **Query Engine** and their interactions with the **Knowledge Base**.

When a question is received, the model processes it through the different layers before letting the **Query Engine** generate an answer. This answer, built upon the **Knowledge Base**, is then processed again by each of the layers in the reversed order. This mechanism is illustrated in Algorithm 1.

Algorithm 1 Algorithm representing the Agent's Cycle

```
1: procedure AgentCyle(question)
2:     UpdateTheoryOfMind(question)
3:     strategy ← SelectStrategy(question)
4:     AdjustStory(question, strategy)
5:     answer ← QueryEngine.Process(question)
6:     FilterAnswer(answer, strategy)
7:     UpdateTheoryOfMind(answer)
8:     return answer
9: end procedure
```

An answer generated by the agent, a *query result*, has the following structure:

$$\langle query, results, extra \rangle$$

If the question was of type *validation*, its result only contains a value, either negative or positive. On the other hand, if it is of type *information-gathering*, the answer contains a list with all the entities and their cardinality, $\langle entity, cardinality \rangle$. The cardinality value represents the number of times an entity appeared in the events that satisfy the *query conditions*. An answer may also contain some additional details in the *extra* field. They are organized in a collection of pairs $\langle property, value \rangle$ and are generated during the agent's cycle. This field can be used by the system responsible of presenting the answer to the user, for instance to generate Natural Language or to animate the agent's avatar.

To the question "When was Peter at Home?", the agent answers with the *query result* with the *query* representing the question, a list with two elements, each one containing a *Time* value, and the *extra* information field is empty.

The following sections elaborate on the functionality each layer and their components have.

3.1 Knowledge Base

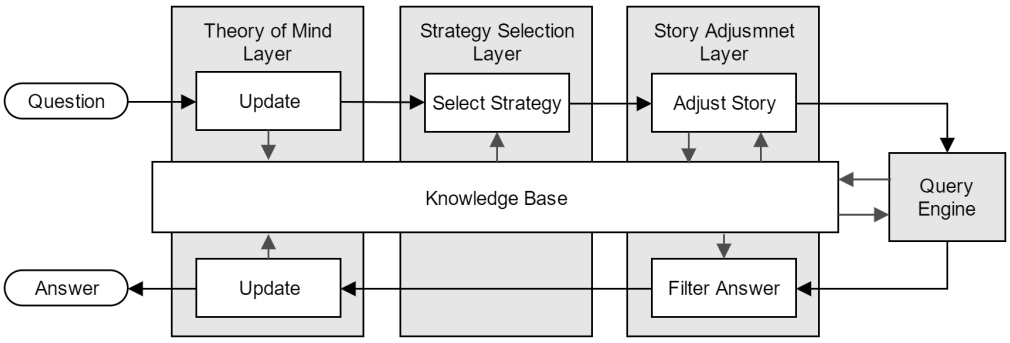

Figure 1: Deceptive Virtual Agent's two pass control layered architecture

The **Knowledge Base** manages the different stories of the model, real and parallels, along with the representation of the interlocutors' knowledge.

The real story represents the original version of the agent's past actions and it won't change during the interaction. The parallel stories are alternative versions of the original story, one for each of the interlocutors the agent is interacting with. Unlike the real story, these stories can change: new events will be created, replaced and partially changed, endowing the agent using our model the capability to adapt the events shared with its interlocutors.

The interlocutors' knowledge is accessed by the **Theory of Mind Layer**. Based on the questions asked and the answers generated, this layer registers the agent's beliefs about what each interlocutor knows about the real story and its particular alternative story. To accommodate this functionality, the **Knowledge Base** has a registry about the story fragments that are known by every interlocutor. This information is stored in a table associating each interlocutor with the its known story's entities and respective events. The role of the **Theory of Mind Layer** is explained in more details in the following section.

3.2 Theory of Mind Layer

The lie creation and maintenance process depends massively on the information the environment external to the agent, in particular its interlocutor, has about the real events. This is known as the external consistency [9]. To improve the agent's chances of deceiving successfully, this layer updates the **Knowledge Base** with the agent's beliefs about the information the others know from the questions asked and the answers shared.

On the first pass, the **Theory of Mind Layer** receives a question structured as a *query*. The layer selects the events that satisfy the *question conditions* and then identifies which entities are in common between these events and the *question conditions*. The layer updates the beliefs related to the interlocutor that asked the question within the **Knowledge Base** with these retrieved entities. However, this Theory of Mind mechanism naively assumes that the interlocutors only ask questions about events and entities they know. This assumption creates room for the use of deceptive strategies from the user, such as questioning the agent about events that are not known.

When the generated answer is going through the **Theory of Mind Layer**, before being shared to the interlocutor, the layer updates the **Knowledge Base** in order to tag which entities will now be known by the interlocutor. By updating its beliefs with the answer's content, in future adjustments, the model will consider this information while generating new lies, such that they are consistent with the user's knowledge.

3.3 Strategy Selection Layer

Using the question asked, the representation of others' knowledge and the sensitive value of each event and entities, this layer will select the best strategy for the current state of the interaction. The procedure SELECT STRATEGY evaluates the environment state, decides if the agent should deceive the interviewer and, if so, chooses the method to apply (See Table 1). The first strategy is the *Don't Lie* strategy, which can be followed by two different methods, *None* and *Hide*.

Strategy	Method
Lie	Duplicate Event Adjust Entity Adjust Event
Don't Lie	None Hide

Table 1: Lie strategies and methods

If the events satisfying the *question conditions* can be adjusted, the agent does not believe the interlocutor knows the whole event and there are alternative entities to replace the compromising ones, the *Lie* strategy is selected. Otherwise, if the story shared with the user cannot be adapted, the chosen strategy is *Don't Lie*.

The strategy *Lie* can be followed by three different methods. If the events that satisfy the query are sensitive and another variant of the story has not been created yet, the selected method is *Duplicate Event*. When applying this method, the mechanism creates a copy of the sensitive events and, for each one of them, alter the sensitive entities that are unknown to the interlocutor (See Algorithm 2). The *Adjust Event* method is used when the event that matches the question is already a duplicated event. Since it is already an altered version of the truth, it adjusts the entity's fields that are in disagreement with the interlocutor's knowledge gathered since the duplication of the event. To apply this method, the model only changes the relevant fields with the result of the procedure SIMILARENTITY, described in section 3.4. When the question focuses on a single sensitive entity that has a large scope, i.e. it appears in multiple events across the whole story, the selected method is the *Adjust*

Entity. To apply this method, the model searches all the occurrences of this entity and replaces it with the entity returned from the procedure SIMILARENTITY. To guarantee the internal consistency of the story [9], the events must not have conflicts between them. Therefore, when there is a chance of changing an entity throughout multiple sensitive events, the model prioritizes this method over the others.

Algorithm 2 Cycle used on Duplicate Event method

```
1: procedure DUPLICATEEVENT(question)
2:     events ← FilterStory(question)
3:     for each event ∈ events do
4:         eventCopy ← Copy(event)
5:         for each field ∈ eventCopy do
6:             if IsSensitiveAndUnkown(field) then
7:                 field ← SIMILARENTITY(field)
8:             end if
9:         end for
10:    end for
11: end procedure
```

The strategy *Don't Lie* has two available methods. The first one is the method *None*, which is applied when the information satisfying the query is safe to share, not sensitive. This causes no alteration in neither the story nor the answer. However, the alternative for the strategy *Don't Lie*, the *Hide* method, is selected when the information shared in the answer is compromising and there is not an alternative version of the story to be created. This approach removes from the answer entities that are sensitive and should be kept hidden before sharing them with the user.

3.4 Story Adjustment Layer

This layer has two roles. Its first role is to use the *question* and the *strategy* received from the previous layer to autonomously adjust the narrative in the **Knowledge Base**, by creating new events or modifying already created ones. Its second role is to modify the answer, generated by the **Query Engine**, according to the previously selected strategy. The three methods associated with the strategy *Lie* create and adjust new events or entities in the parallel story, and they are applied when this layer handles the question by the procedure ADJUSTSTORY, before it is processed by the **Query Engine**. On the other hand, the procedure FILTERANSWER is in charge of applying the method *Hide* after the answer is generated.

In this layer, when the strategy selected changes the story's content, the model needs to replace sensitive entities with other ones. The procedure SIMILARENTITY is used to find a suitable replacement for an entity that should not be shared. Our model searches on its **Knowledge Base** for possible alternatives and ranks the candidates based on three heuristics: how sensitive they are, how similar their context is with the original one, and if the agent believes the interlocutor already knows them.

3.5 Query Engine

Since our dialogue is a question-answering interaction, we used a query system to build the core of our mechanism. The interlocutor's questions follow the structure previously described and the agent, by fetching the corresponding results within its **Knowledge Base**, returns the events and entities that match the question's conditions from the story shared with that interlocutor. In order to maximize the relevance of the information shared, an event is included in an answer only if it satisfies all the question's conditions.

4. INTERROGATION GAME

The environment used to test the performance of our autonomous agent's model is an Interrogation Game where the player acts as a police officer and a virtual agent equipped with our model acts as a criminal suspect.

In this game, the interlocutor interview the suspect in order to validate the information he/she already has about the investigation and discover new facts related with the case. Whereas our Virtual Suspect was a deceptive agent, the player were not told this particular feature of the character. To create a convincing experience, we designed the interactions following what happens in real interviews. We studied police protocols, procedures and the interaction's characteristics specific to this context to understand how an interrogation is usually conducted.

4.1 Police Interview

Across the world, law enforcement agencies from different countries can follow different practices and protocols during an investigation [12], specifically during interviews [13]. However, some techniques are commonly used by all of them. For instance, it is established that a police officer must not go empty handed to an interview. He/She has to be aware of the information already in the possession of the police such as the information gathered from physical evidences or from the interviewee's criminal record. With these information, the police officer prepares for the interview a list of facts and notes relevant to the case as well as a question plan to guide him/her during the conversation.

The interview's preparation is an important phase of an investigation but the actual interrogation follows specific protocols to guarantee the best outcome from the conversation. Ideally, the police officer aims to discover new information about the case and expose any inconsistency from the interviewee's statements. In order to achieve this objective, the interviewer is taught to ask open and generic questions in the beginning of the conversation, also called the information gathering phase, and only during the final phase of the interrogation the police officier should try to confront the suspect with information he/she already has [22]. This method is commonly used by police forces to let an interviewee expose contradictory information with which they can pressure the suspect to, hopefully, reveal compromising details in his/her statements.

Even if a police officer extracts compromising information from the suspect, to be considered admissible in court, the methods used to acquire this information must satisfy the laws of the country. During an information exchange with our country's Police, we have been told that to guarantee that the result of a interview can be used in court, their investigators can not say false information or show fake evidence to the suspect.

Taking into account these protocols and best practices, we developed the Interrogation Game, in which players were able to experiment with these procedures.

4.2 Implementation

The agent's main component is the query system (see section 3.5) and all the other functionalities are built around it

using a modular architecture. Each of the layers have two handlers, one for processing the question and another one for processing the answer. This approach allows the model to be evolutive as new modules,i.e. layers, can be developed and stacked with the existing ones.

We coupled the implementation of the Knowledge Base, the component in charge of storing the story, events and entities, with the development of an API allowing to easily create, retrieve, modify and search its content. To help scenario's authors, we developed a set of tools to load a story from an XML file into the model. We also implemented a simple template based natural language generator to create readable answers based on the queries' result.

Our model was configured to play the role of a suspect in an interview. Since we built this system as a game, we faced some game design challenges. In particular, we realized that if the agent were to use the *Hide* strategy, it could prevent the player to explore the narratives of the suspect because of the limited set of actions at his disposal. In some cases, the best strategy for the agent could be to hide information but in order to not frustrate the player with empty answers, we decided to remove it. In a future work, we plan to implement our model in a different context that ensures that all the strategies are exploitable.

As we mentioned before, our approach let the agent thinks that everything the interlocutor says is true. Again, this an interesting development for future work in order to develop additional strategies to assess the veracity of a statement. However, in the context of an Interrogation Game, it fits well as police officers should not use false information to ensure the validity of the interrogation. Therefore, our agent will tag all information coming from the player as *Known*.

4.3 Answer Generation

Using our model, the Deceptive Virtual Suspect outputs its answers as structured messages, i.e. a *query result*. We developed a simple template based natural language generation mechanism which translates these structured messages to sentences understandable by the player.

This system processes the generated answer based on its question's type. The validation type question only had two possible values, positive or negative, therefore, the generated sentences for each answer were respectively "Yes." or "No.". Regarding the information gathering questions, we had to develop a solution to handle the different types of answers.

To generate sentences from answers with multiple entities, our mechanism uses the following template:

$$< \textit{subject action entities} >$$

When the Deceptive Virtual Suspect is the only agent involved on the action, the *subject* is replaced by "I" and, if more agents are involved, it is replaced by "we". The field *action* contains the past tense form of the verb described in the event's action field. The last field is a string with the entities listed within the answer with their respective cardinality. If the entities contained in the answer are of type *Time*, multiple values with the same day are grouped together.

4.4 Game Interaction

The Interrogation Game followed a turn based approach: the player asks a question to the Deceptive Virtual Suspect which answers it. Then, the player can ask another question and so forth.

In order to build the interface of the game, we conducted several user studies with a focus group aiming to test its usability. The feedback gathered helped us reach the final version of the interface.

This interface is composed of the following elements. First, the player has access to a list of predetermined questions he can ask to the Deceptive Virtual Suspect. Then, a notebook is present to guide the player during the Interrogation Game. It contained details and notes about the investigation gathered before the interview and can be accessed by the player at any point during the interaction. Finally, a textual log is shown to the player with all the questions asked to the Deceptive Virtual Suspect during the game and the associated answers. The player is free to interact with the Deceptive Virtual Suspect as much as he wants and once he is satisfied with the interview's result, when he thinks he obtained the information he needed, the player can end the interaction.

4.5 Authoring

By itself, the Deceptive Virtual Suspect is capable of adjusting a story during an interaction, however it requires a original story from which it will build the alternative ones. Therefore, an author needs to create a story and then load it into the Deceptive Virtual Suspect's memory. When formalizing the story, he needs to specify the entities the story has, associate them into events and assign an incriminatory value to these events. Then, the story can be loaded into the virtual suspect's memory and the author can test its behaviour. If he thinks the generated lies of the agent are following a unintended story line, he can add events and entities that are not real but will influence our lying mechanisms and, consequently, the generated plots.

To evaluate the Deceptive Virtual Suspect capabilities with players, we designed a narrative in which the Deceptive Virtual Suspect endorse the role of the main character named Peter Barker. Peter Barker is the only suspect of a jewelry shop robbery. A silver necklace was stolen and the police established several connections between him and the man trying to sell the stolen good. The narrative was translated to the story representation format used in the model and preliminary tests were conducted with a focus group. We found that the narrative offered a good challenge for the users and, at the same time, allowed us to test the agent's deceiving capabilities.

5. DEMONSTRATION

This section demonstrates the virtual agent's model behaviour, implemented within the Deceptive Virtual Suspect. We are going step by step, describing the values of each variables, the content of the messages and explaining how each layers intervene in the answering process of the Deceptive Virtual Suspect.

For instance, lets consider that a player asks the question "Where were you on March 5th between 4:00 and 4:30?" to the Deceptive Virtual Suspect. The **Theory of Mind Layer** handles it first. This layer detects that the interviewer knows about an event that occurs between 4:00 and 4:30 on March 5th and that "Peter Barker", the Deceptive Virtual Suspect's character, is involved. With the beliefs about the user's knowledge updated, the **Strategy Selector Layer** identifies the events that match the query's conditions and verifies that they are incriminatory. After ver-

ifying that there are no false events created to conceal the ones identified and that the Deceptive Virtual Suspect can autonomously replace them with less compromising events, this layer suggests the strategy *Lie* using the method *Duplicate Event*.

To apply this method, the **Story Adjustment Layer** duplicates the event, copies the known fields (Time and Agent) and replaces incriminatory fields (See Figure 2). To replace the incriminatory entities with less compromising ones, this layer invokes SimilarEntity to get an ordered list of replacements. This procedure is invoked to replace the entities in the fields *Location* and *Theme*. Since there are other events that occurred on the same Time Period, they also involved "Peter Barker" and they are not compromising, their entities have an higher rank than others. So, the **Story Adjustment Layer** selects them to replace the compromising entities on the event duplicated.

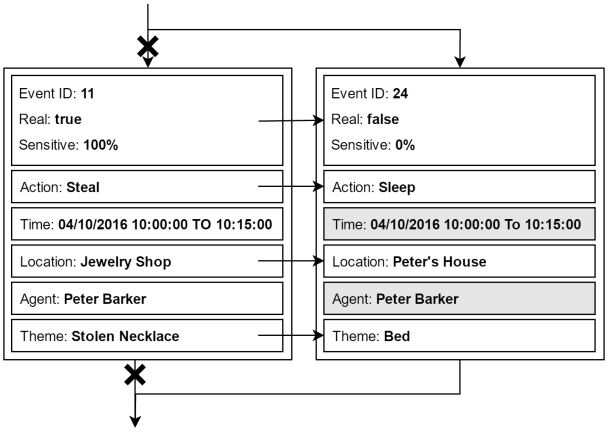

Figure 2: Event Duplication

After adjusting the story to replace the compromising events with false non-incriminatory events, the **Query Engine** retrieves them instead of the real ones. It processes the satisfied entities, the entities associated with the *Location* field of the previously selected events, and generates the respective answer.

The query result now goes through the three layers in a reverse order, and in this example, only the **Theory of Mind Layer** uses it. Based on the entities shared on its content, this layer tags them as known in the **Knowledge Base**. Finally, the answer created by the Deceptive Virtual Suspect is sent to the player.

6. EVALUATION

We aimed to evaluate how the users perceived the Deceptive Virtual Suspect's answers during the Interrogation Game and how different questions' sequences would impact the agent's lie generation.

We designed an experiment where participants would interact with the Deceptive Virtual Suspect through the Interrogation Game. We recruited 31 participants (24 male, 7 female) aged between 18 and 30 years old. Each session had a duration of around ten minutes in which the participant had to interview the Deceptive Virtual Suspect, Peter Barker, in order to validate a set of information that was previously made available to him.

From each session, we collected the sequence of questions asked by the participant, the answers generated by the Deceptive Virtual Suspect, the agent's memory state throughout the interaction and the result of a questionnaire about the participants experience in interrogation, the knowledge of the story he acquired after the interview and his confidence regarding the Deceptive Virtual Suspect's answers.

From the questionnaires, we observed that none of the 31 participants had previous training on interviewing techniques and most of them were unfamiliar with police interview methods.

Regarding the questions about their understanding of the story, we also asked them to indicate the degree of confidence towards their answers. The response of the participants are summarized in Table 2.

Questions	Answer		Confidence	
	Yes	No	Mean	Std. Deviation
Did the Virtual Suspect sell the stolen necklace?	71%	29%	3,30	1,013
Did the Virtual Suspect rob the jewelry store?	42%	58%	3,30	1,279
Is the Virtual Suspect guilty of the accusations?	87%	13%	3,45	1,028

Table 2: Descriptive analyze of participants' answers about the case and their confidence

When questioned about the frequency of lies shared by the suspect during the interaction, 87,1% of the participants believed that the Deceptive Virtual Suspect expressed a mix of lies and true statements, while 12,9% thought it constantly lied.

83,9% of the participants asked a sequence of questions that was able to expose, at least once, the Deceptive Virtual Suspect lies by contradicting its answer with information offered to the users in their notebook. The sequence of questions produced during the interaction by these participants were grouped in G_0 and the remaining sequences on G_1. Then, we grouped the Deceptive Virtual Suspect's memory logs, based on G_0 and G_1, and retrieved for each sequence the number of false events that have been created and adjusted, as well as the number of compromising real events that have been shared. The results are depicted in Table 3

Question Sequences		G_0	G_1
Percentage		83,9%	16,1%
Number of False Events Created	Mean	6,35	5,80
	Std. Deviation	0,689	0,837
Number of False Events Adjusted	Mean	1,27	0,60
	Std. Deviation	0,604	0,548
Number of Real Events Shared	Mean	3,81	0,20
	Std. Deviation	0,634	0,447

Table 3: Descriptive analysis of the Deceptive Virtual Suspect's memory state grouped by questions' sequence quality

In order to explore how the lying strategies triggered within the agent were impacted by the strategies a person uses dur-

ing an interrogation, we conducted one way ANOVA tests. We defined as independent variable the group (G_0 or G_1) and as dependent variables the number of compromising real events shared, the number of false events created and the number of false events adjusted. The results showed that there was a statistically significant difference of real events shared ($F(1,29) = 146.04; p < 0.005$) and false events adjusted ($F(1,29) = 5.28; p = 0.029$) between the two groups. However, no significant difference was found for the number of false events created ($F(1,29) = 2.47; p = 0.127$). In other words, while the number of created lies remained similar in both groups, it would appear that when the interrogator follows a strategy of asking open-questions first, it led the Deceptive Virtual Suspect to adjust his story multiple times and to share real information, resulting in the lies being exposed.

6.1 Discussion

The results of the questionnaires showed that all the participants believed after the interaction that the Deceptive Virtual Suspect was lying. The context might have led the participants to assume the agent was lying while filling out the questionnaires but it is still important to note that no participant assumed that the suspect was to be trusted.

Additionally, when asking the participants how confidently they thought their choice of questions influenced the Deceptive Virtual Suspect's answers, using a 5-point Likert scale, they answered with a low confidence, with a mean value of 2.81 and standard deviation of 0.910. Because they had to choose from a set of predefined question, the participant might felt that they did not have a strong impact in the interview. Future improvements on how the player asks questions to the agent could improve his game experience and create more diversified sequences of questions to evaluate the Deceptive Virtual Suspect's performance.

When looking at the generated sequences of questions and answers, we observed that the sequences following the best practices in police methodologies, i.e. open questions first followed by closed questions, made the Deceptive Virtual Suspect readjust its shared story multiple times. This lead the agent to reveal compromising information about real events during the interaction. On the other hand, a sequence of questions that did not follow the police protocol's rules not only could not expose as many lies (94.8% less real events were shared), but also made the Deceptive Virtual Suspect adjust its story less times when compared to the other set of questions' sequences (52.76% less events were readjusted). This is a promising observation as this indicates that our model could produce lies and eventually contradictory statements in a similar fashion as an actual suspect. An actual evaluation and validation of this process will be required in future steps of our research.

7. CONCLUSION

In this article we presented a virtual agent's model aimed at endowing a virtual character with the capability to autonomously generate alternatives in its personal story in order to deceive and mislead its interlocutors about past events. The proposed model reduces the effort of a story's author since he does not need to write alternative stories. The model can analyse the events and the entities stored in the agent's memory and, depending on their sensitivity, it can dynamically generate and adjust false alternatives to share with its interlocutors. An implementation of the model has been realized within an interrogation game environment and a preliminary experiment was conducted to assess the potential of such a system.

The evaluation results revealed that the behaviour of the agent showed significant differences depending on the interrogation methodologies followed by the players of the Interrogation Game, event though these participants seemed to feel that they had no impact on the behaviour of the agent. When the players asked open questions before closed questions, it led the agent to readjust its false stories multiple times and to share more real information. This result can suggest that our model could represent a helpful tool for training and teaching specific interviewing skills to police trainees. However, despite showing promising results, the scenario of our experiment limited the interaction to a predetermined set of questions aimed at one suspect. Another scenario for the Interrogation Game was prepared, containing an intricate and complex narrative with multiple suspects and witnesses. However, despite the model being designed to support multi-agent scenarios, we chose for this first experiment to focus on the interrogation of one suspect instead.

The perspectives for our work do not only include aiming at evaluating the model in a richer multi-agent environment, but also exploring how to extend and improve it. Its architecture allows new layers of behaviours to be added easily and future improvements could include investigating how to automatically compute the sensitive value of the entities based on social factors such as emotions or interpersonal relations. Additionally, we could also investigate how higher levels of Theory of Mind could impact the agent's strategies and how communicative modalities (speech and non-verbal behaviour) could be used by the agent in this deceptive process. In the future, we aim to address the discussed limits and improve the model as, although we demonstrated how a deceptive agent could be used in the context of police interviews and Human-Agent Interaction (through an Interrogation Game), we believe that such an agent could find application in other areas like emergent narrative generation, NPC's in video-games or Human-Robot Interaction.

REFERENCES

[1] M. Bruijnes, J. Kolkmeier, J. Linssen, M. Theune, D. Heylen, et al. Keeping up stories: design considerations for a police interview training game. 2013.

[2] M. Bruijnes, S. Wapperom, D. Heylen, et al. A virtual suspect agent's response model. 2014.

[3] J. Campos and A. Paiva. May: My memories are yours. In *International Conference on Intelligent Virtual Agents*, pages 406–412. Springer, 2010.

[4] C. Castelfranchi, R. Falcone, and F. De Rosis. Deceiving in golem: how to strategically pilfer help. In *Autonomous Agent'98: Working notes of the Workshop on Deception, Fraud and Trust in Agent Societies*. Citeseer, 1998.

[5] M. A. Conway. Sensory–perceptual episodic memory and its context: Autobiographical memory. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 356(1413):1375–1384, 2001.

[6] B. M. DePaulo, M. E. Ansfield, S. E. Kirkendol, and

J. M. Boden. Serious lies. *Basic and applied social psychology*, 26(2-3):147–167, 2004.

[7] B. M. DePaulo and D. A. Kashy. Everyday lies in close and casual relationships. *Journal of personality and social psychology*, 74(1):63, 1998.

[8] J. Dias, W. C. Ho, T. Vogt, N. Beeckman, A. Paiva, and E. André. I know what i did last summer: Autobiographic memory in synthetic characters. In *International Conference on Affective Computing and Intelligent Interaction*, pages 606–617. Springer, 2007.

[9] V. A. Gombos. The cognition of deception: the role of executive processes in producing lies. *Genetic, Social, and General Psychology Monographs*, 132(3):197–214, 2006.

[10] J. Gratch, Z. Nazari, and E. Johnson. The misrepresentation game: How to win at negotiation while seeming like a nice guy. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 728–737. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[11] J. P. Hespanha, Y. Ateskan, and H. Kizilocak. Deception in non-cooperative games with partial information. In *Proceedings of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control*. Citeseer, 2000.

[12] Y. Ma. A comparative view of the law of interrogation. *International Criminal Justice Review*, 17(1):5–26, 2007.

[13] C. Meissner, A. Redlich, S. Bhatt, and S. Brandon. Interview and interrogation methods and their effects on investigative outcomes. *Campbell systematic reviews*, 8(13), 2012.

[14] R. Quirk, S. Greenbaum, G. Leech, J. Svartvik, and D. Crystal. *A comprehensive grammar of the English language*, volume 397. Cambridge Univ Press, 1985.

[15] M. Riedl, C. J. Saretto, and R. M. Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 741–748. ACM, 2003.

[16] M. Riedl, D. Thue, and V. Bulitko. Game ai as storytelling. In *Artificial Intelligence for Computer Games*, pages 125–150. Springer, 2011.

[17] A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[18] E. Tulving. Episodic and semantic memory 1. *Organization of Memory. London: Academic*, 381(4):382–404, 1972.

[19] E. Tulving. What is episodic memory? *Current Directions in Psychological Science*, 2(3):67–70, 1993.

[20] A. Vrij, K. Edward, K. P. Roberts, and R. Bull. Detecting deceit via analysis of verbal and nonverbal behavior. *Journal of Nonverbal Behavior*, 24(4):239–263, 2000.

[21] A. Vrij, R. Fisher, S. Mann, S. Leal, B. Milne, S. Savage, and T. Williamson. Increasing cognitive load in interviews to detect deceit. *International developments in investigative interviewing*, pages 176–189, 2009.

[22] A. Vrij, S. Mann, and R. P. Fisher. Information-gathering vs accusatory interview style: Individual differences in respondents' experiences. *Personality and individual differences*, 41(4):589–599, 2006.

[23] J. J. Walczyk, L. L. Harris, T. K. Duck, and D. Mulay. A social-cognitive framework for understanding serious lies: Activation-decision-construction-action theory. *New Ideas in Psychology*, 34:22–36, 2014.

[24] J. J. Walczyk, K. S. Roper, E. Seemann, and A. M. Humphrey. Cognitive mechanisms underlying lying to questions: Response time as a cue to deception. *Applied Cognitive Psychology*, 17(7):755–774, 2003.

[25] E. G. Weber. *Varieties of questions in English conversation*, volume 3. John Benjamins Publishing, 1993.

[26] M. Zuckerman, B. M. DePaulo, and R. Rosenthal. Verbal and nonverbal communication of deception. *Advances in experimental social psychology*, 14(1):59, 1981.