Explaining and Predicting the Behavior of BDI-Based Agents in Role-Playing Games*

M.P. Sindlar, M.M. Dastani, F. Dignum, and J.-J.Ch. Meyer
{michal,mehdi,dignum,jj}@cs.uu.nl

University of Utrecht
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

Abstract. Virtual characters in games operate in a social context involving other characters and human players. If such socially situated virtual characters are to be considered believable, they should be able to adjust their behavior based on beliefs about the mental states of other characters and human players. Autonomous BDI-based agents are suitable for modeling characters that exhibit such intentional behavior. In this paper, it is illustrated how agent-based characters can infer the mental state of other players by observing those other players' actions in the context of a declarative game specification. The game specification can be utilized in explanation and prediction of agents' behavior, and as such can form the basis for developing socially aware characters.

1 Introduction

For games and simulations with interactive virtual characters to provide users with a satisfying experience, it is of vital importance that those characters are believable to the user. Appearing to pursue goals and to be responsive to social context are determining factors for believability [1], and interaction with virtual characters is richer and more enjoyable if these anticipate the behavior of other characters [2]. Believable characters that operate in a social context should exhibit social awareness and not only pursue their own interests, but also be able to take the mental states of other characters into account if they believe these to conflict or coincide with their own goals and beliefs.

Statistical approaches to game-based plan recognition exist [3] but require large amounts of gameplay data to be processed, which might not always be available. Recent work in the agent programming community has focused on recognizing an agent's plan on grounds of its program and observed actions, and inferring the mental state of the agent that plausibly explains observed behavior [4,5]. This offers promising directions for developing socially aware virtual characters, as characters which can infer other characters' mental states have the possibility to incorporate attributed mental states into their own decision-making

* This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

process. This allows for plausibly misguided behavior and thus contributes to believability [6]. However, current approaches ignore the setting in which agents operate and inferred explanations are independent of social context. If characters in games are designed to behave as autonomous agents, then a game can be regarded as an agent society [7]. The behavior of characters in such a game takes place in the context of the society, which can be captured and specified by means of an organizational model [8].

This paper gives a declarative solution to abducing the mental state of virtual characters implemented as BDI-based agents, which takes into account the context in which these characters operate. In Sect. 2 mental state abduction is reviewed. Sect. 3 introduces the game-related context in terms of scenes and roles, and in Sect. 4 it is shown how this context can be utilized in explaining and predicting agent behavior. Sect. 5 ties things together in an example, and Sect. 6 concludes with a brief discussion and ideas for future research.

2 Mental State Abduction

In this section it is described how the observed behavior of agents can be related to an explanation in terms of a description of their mental state, recapitulating our work in [5]. In Defs. 1–2 the behavior of agents is described, and in the remainder of this section it is shown how behavior is observed and explained.

Let $\mathcal{L}$ be a propositional domain language with negation and conjunction over propositions. $\mathsf{Lit} \in \mathcal{L}$ is the set of literals in this language, and $\mathcal{L}_\Gamma \subseteq \mathcal{L}$ a simple language allowing only (conjuncted) literals. Let Act be a set of action names. The behavior of an agent can be described as an expression consisting of actions, which are either atomic observable actions $\alpha \in \mathsf{Act}$, or tests $\phi?$ on propositions $\phi \in \mathcal{L}$. Actions can be composed by means of operators for sequential composition (;) and choice (+).

Definition 1 (behavioral description) *Let $\alpha \in \mathsf{Act}$ be an atomic observable action, and $\phi?$ the test action on proposition $\phi \in \mathcal{L}$. The set of behavioral descriptions $\mathcal{L}_\Pi$ with typical element π is then defined as follows.*

$$\pi ::= \alpha \mid \phi? \mid \pi_1;\pi_2 \mid \pi_1 + \pi_2$$

Note that there is no notion of iteration in $\mathcal{L}_\Pi$. It is assumed, though, that behavior can be iteratively performed if an agent reapplies a behavioral rule of the type defined in Def. 2. Such a rule states that the behavior described by π is appropriate for achieving the goal γ if the condition β is believed to hold, and is taken to be interpreted in the operational context of an agent program [9,10].

Definition 2 (behavioral rules) *Behavioral rules specify a relation between behavior $\pi \in \mathcal{L}_\Pi$ and a mental state consisting of achievement goals $\gamma \in \mathcal{L}_\Gamma$ and beliefs $\beta \in \mathcal{L}$. The set of behavioral rules $\mathcal{L}_{\mathcal{BR}}$ has br as its typical element.*

$$\mathsf{br} ::= \gamma \leftarrow \beta \uparrow \pi$$

The atomic actions $\alpha \in \mathsf{Act}$ of an agent are taken to be publicly observable, such that a perceived action can be directly related to the performed action. Because behavioral descriptions do not allow for concurrent actions and interpretation of the agent program is assumed to do so neither, the perception of multiple actions performed by a single agent is taken to be sequential. To distinguish sequential perception of actions — which is an incremental process — from actions in sequence as part of a plan, percepts are presented as a list.

Definition 3 (percepts) *Let $\alpha \in \mathsf{Act}$ be an observable action and ϵ a special empty (null) action, such that $(\epsilon\delta) = (\delta) = (\delta\epsilon)$. The set of percept expressions $\mathcal{L}_{\Delta}$, with typical element δ, is then defined as follows.*

$$\delta ::= \ \alpha \mid \epsilon \mid \delta_1\delta_2$$

In order to explain perceived actions in intentional terms, a relation has to be established between perceived actions, descriptions of behavior, and the behavioral rules that connect a mental state (goals and beliefs) to behavior. This relation should be defeasible, because on grounds of observed actions alone it is not necessarily possible to analytically infer the mental state that caused the agent to perform those actions. For this reason the behavioral rules of Def. 2 are described as the logical implications defined in Def. 4, which state that a precondition consisting of a goal and belief description implies a certain behavior.

Definition 4 (rule description) *Let $(\gamma \leftarrow \beta \uparrow \pi) \in \mathcal{L}_{\mathcal{BR}}$ be a behavioral rule. The set of rule descriptions $\mathcal{L}_{\mathcal{RD}}$, with typical element rd, is defined as follows.*

$$\mathsf{rd} ::= \ \mathsf{goal}(\gamma) \wedge \mathsf{belief}(\beta) \Rightarrow \mathsf{behavior}(\pi)$$

The function $\mathsf{desc} : \mathcal{L}_{\mathcal{BR}} \longrightarrow \mathcal{L}_{\mathcal{RD}}$ maps rules to their description, such that $\mathsf{desc}(\mathsf{br}) = (\mathsf{goal}(\gamma) \wedge \mathsf{belief}(\beta) \Rightarrow \mathsf{behavior}(\pi))$ for any $\mathsf{br} = (\gamma \leftarrow \beta \uparrow \pi) \in \mathcal{L}_{\mathcal{BR}}$.

Because behavioral rules, as defined in Def. 2, are interpreted in an operational context, the implications in the rule descriptions defined in Def. 4 do not hold in a classical logical way with respect to describing agent operation. However, if perceived actions $\delta \in \mathcal{L}_{\Delta}$ can be related to the behavioral descriptions $\pi \in \mathcal{L}_{\Pi}$, then logical abduction — which states that on grounds of an observation ψ and a logical implication $\phi \Rightarrow \psi$, the defeasible explanation ϕ can be abduced — can be used to infer the preconditions of logical rule descriptions. To relate action sequences to descriptions of behavior, a function is defined that maps behavioral descriptions to sets of *observable traces* of behavior by filtering out internal tests, which are taken to be unobservable, and branching observable traces at points where choice occurs. The operator $\cup$ is standard set union, and $\circ : \wp(\mathcal{L}_{\Delta}) \times \wp(\mathcal{L}_{\Delta}) \longrightarrow \wp(\mathcal{L}_{\Delta})$ is a non-commutative composition operator defined as $\Delta_1 \circ \Delta_2 = \{\ \delta_1\delta_2 \mid \delta_1 \in \Delta_1 \text{ and } \delta_2 \in \Delta_2\ \}$, where $\Delta_1, \Delta_2 \subseteq \mathcal{L}_{\Delta}$.

Definition 5 (observable trace function) *Let $\alpha \in \mathsf{Act}$, $\phi \in \mathcal{L}$ and $\pi \in \mathcal{L}_{\Pi}$. The function $\tau : \mathcal{L}_{\Pi} \longrightarrow \wp(\mathcal{L}_{\Delta})$ is then defined as follows.*

$$\begin{array}{lll} \tau(\alpha) & = & \{\alpha\} \\ \tau(\phi?) & = & \{\epsilon\} \end{array} \qquad \begin{array}{lll} \tau(\pi_1 + \pi_2) & = & \tau(\pi_1) \cup \tau(\pi_2) \\ \tau(\pi_1; \pi_2) & = & \tau(\pi_1) \circ \tau(\pi_2) \end{array}$$

In order to abduce mental state preconditions of the logical rule descriptions in Def. 4 on grounds of observed actions, a relation must be established between an observed action sequence and the traces that represent the observable aspect of the behavior described in the behavioral description part of a rule. If every action of the agent is observed and the rules completely describe an agent's possible behavior, then an observed sequence of actions can be related to the (non-strict) prefix of an observable trace of behavior described by some $\pi \in \mathcal{L}_{\Pi}$.

Definition 6 (structural relations) *Let $\preccurlyeq \subseteq \mathcal{L}_{\Delta} \times \mathcal{L}_{\Delta}$ be the prefix relation on sequences $\delta, \delta' \in \mathcal{L}_{\Delta}$ and $\succcurlyeq \subseteq \mathcal{L}_{\Delta} \times \mathcal{L}_{\Delta}$ the suffix relation, defined as follows.*

$$\delta \preccurlyeq \delta' \text{ iff } \exists \delta'' \in \mathcal{L}_{\Delta} : \ (\delta' = \delta\delta'') \qquad \delta \succcurlyeq \delta' \text{ iff } \exists \delta'' \in \mathcal{L}_{\Delta} : \ (\delta' = \delta''\delta)$$

Note that every δ is a prefix and suffix of itself iff δ'' is the empty action ϵ.

Defining different structural relations, as shown in [5], may allow for relating observed actions to observable traces also in the case that not every action is observed. It was proven that this leads to an increase in abduced explanations, and that the set of explanations inferred on grounds of complete observation is a subset of explanations in case of partial observation. In order to focus on the way contextual information can be used to facilitate the process of mental state abduction, it is assumed here that observation is complete.

An agent's behavior is to be explained in terms of a description of its mental state. In order to refer to these preconditions, let the set $\mathcal{L}_{\Omega}$ with typical element ω be defined as $\mathcal{L}_{\Omega} = \{ \ \mathsf{goal}(\gamma) \wedge \mathsf{belief}(\beta) \mid \gamma \in \mathcal{L}_{\Gamma}, \beta \in \mathcal{L} \ \}$. An explicit 'lifting' notation for functions is used, such that for any $\mathfrak{f} : D \longrightarrow D'$, the *lifted* version of $\mathfrak{f}$ is ${}^{\wp}\mathfrak{f} : \wp(D) \longrightarrow \wp(D')$, such that for $\Phi \subseteq D$, ${}^{\wp}\mathfrak{f}(\Phi) = \{ \ \mathfrak{f}(\phi) \mid \phi \in \Phi \ \}$.

An explanatory function is now defined that maps observed action sequences $\delta \in \mathcal{L}_{\Delta}$ to preconditions of rule descriptions of the type $\mathsf{rd} \in \mathcal{L}_{\mathcal{RD}}$, such that δ is a (partial) trace of the behavior 'implied' by the rule description.

Definition 7 (explanatory function) *The function $\chi : \mathcal{L}_{\mathcal{RD}} \longrightarrow \wp(\mathcal{L}_{\Omega} \times \mathcal{L}_{\Delta})$ maps a rule description to a set of tuples of precondition and trace.*

$$\chi(\omega \Rightarrow \mathsf{behavior}(\pi)) = \{ \ (\omega, \delta) \mid \delta \in \tau(\pi) \ \}$$

Let $\delta \in \mathcal{L}_{\Delta}$ be a percept and $\mathcal{RD} \subseteq \mathcal{L}_{\mathcal{RD}}$ a set of rule descriptions. The explanatory function $\mathsf{explain} : \mathcal{L}_{\Delta} \times \wp(\mathcal{L}_{\mathcal{RD}}) \longrightarrow \wp(\mathcal{L}_{\Omega})$ is then defined as follows.

$$\mathsf{explain}(\delta, \mathcal{RD}) = \{ \ \omega \mid \exists (\omega, \delta') \in {}^{\wp}\chi(\mathcal{RD}) : [\delta \preccurlyeq \delta'] \ \}$$

Somewhat less formally, the explanatory function defined in Def. 7 states that the precondition of a rule description is in the set of explanations for a certain observed sequence of actions, if the observed sequence is a (non-strict) prefix of any trace of the behavioral description which is described in the postcondition of the rule description. The function as defined here is not intended to be computationally efficient. It can be proven, though, that $\mathsf{explain}(\delta\delta', \mathcal{RD}) \subseteq \mathsf{explain}(\delta, \mathcal{RD})$ for any $\delta, \delta' \in \mathcal{L}_{\Delta}$, allowing for an efficient implementation.[1]

[1] The authors express their thanks to Henry Prakken for pointing out this stronger and more concise version of their proofs in [5].

3 Agents Playing Games

Agents in a multi-agent system each have their mental state and (inter)act in pursuit of their private goals, taking into account their beliefs and the means provided by their environment(s) [9]. An *agent-based game*, must be more than a regular multi-agent system, as the latter lacks particular qualities that a game might be required to have. When implementing game characters as autonomous agents, a designer gives away part of the behavioral control that a scripting-based approach to character design provides [11]. In return, the daring move of the designer is rewarded with emergent stories that take unexpected turns because of decisions made by the autonomous characters. However, there are certain aspects of the game's 'flow of events' that the game designer wants to ensure, without having to rely on providence or agents' good insight.

In this section declarative game-related concepts are defined, inspired by organizational principles, that are used to illustrate how an agent-based game can be designed that respects some storyline marked out by the designer. Moreover, the same concepts can be used as a guideline with respect to agents' expected behavior, as will be shown in Sect. 4.

3.1 A Declarative Game Specification

The general concept 'game' is hard to define, so that in the present approach a particular kind of game is considered, namely the *agent-based role-playing game*. Such a game is considered to be populated by virtual characters implemented as autonomous BDI-based agents, which play roles similar to the way actors do in a movie. Autonomous agents, however, may be allowed more freedom in the way they enact their role than movie actors are. Such role-enacting agents can be allowed to have private goals that supercede or conflict with those specified by their role, which will show in the behavior they exhibit [12,7].

Most definitions of the concept 'role' recognize that a role comes with obligations, permissions, authority, or the right to perform certain actions. As such, a role describes behavior which can be expected of the role-enacting agent [13]. Roles are therefore defined in a way that encompasses both the descriptive and prescriptive aspect by providing an agent with goals to achieve, information made available to the role-enacting agent, and behavioral rules. These concepts correspond to the B(eliefs), D(esires), and I(ntentions) of the BDI-paradigm, and as such can form the basis for design of the role-enacting agent. Note that the relation of role-derived goals and goal-directed rules is taken to be not necessarily one-to-one; multiple rules for a single goal can exist, or a conjuctive goal may be provided by the role where only rules for literal goals exist. To be able to refer to roles and other entities uniquely, a set of constants ID is introduced.

Definition 8 (role) *Let $\Gamma_{\leq} = (\Gamma, \leq_\Gamma)$ be an ordered set of goals $\Gamma \subseteq \mathcal{L}_\Gamma$, with $\leq_\Gamma \subseteq \mathcal{L}_\Gamma \times \mathcal{L}_\Gamma$ a partial order on Γ. Let $\mathcal{I} \subseteq \mathcal{L}$ be role-accessible information, and $\mathcal{BR} \subseteq \mathcal{L}_{\mathcal{BR}}$ a set of behavioral rules. A role R, identified by a unique identifier $r \in \mathsf{ID}$, is then defined as $\mathsf{R} = \langle r, \Gamma_{\leq}, \mathcal{I}, \mathcal{BR} \rangle$.*

A typical role, featured in many games of the role-playing game (RPG) genre, is that of the *thief*. Unsurprisingly, the thief-role may provide the goal to take possession of a particular item by stealing it. Moreover, a thief could have the goal to steal the item whilst double-checking that nobody is near. If the thief assesses a particular situation to be risky, the goal to steal the item but also ensure that nobody is around might supercede the goal to just steal the item.

Roles do not per definition remain unchanged throughout a game. The context in which a role is enacted influences the way it should be enacted, and this context may change as things happen in the game. Autonomous agents can be given the liberty to enact their role as they see fit, resulting in different types of behavior given the same role specification. Nevertheless, agents are restricted in their actions by the opportunities provided by their environment, and by the norms of the agent society in which they operate. To formalize the norms that regulate behavior, a language of normative expressions $\mathcal{L}_{\mathcal{N}}$ is defined which captures *prima facie norms*, with typical element N, such that $\mathsf{N} ::= F(\alpha) \mid O(\alpha)$. The expression $F(\alpha)$ states that the action $\alpha \in \mathsf{Act}$ is forbidden, $O(\alpha)$ states that the action is obligatory.

In [14], prima facie norms are defined to be *norms [which] usually do not arise from actions, but arise in certain situations [and remain] valid as long as the situation in which they arise stays valid.* Scenes are taken to constitute the norm-governed context in which roles remain unchanged. The scene definition includes the roles figuring in the scene, a set of norms pertaining to the scene, and a set of literals denoting an initial environment state.

Definition 9 (scene) *Let $\mathcal{R}$ be a set of roles as defined in Def. 8, $\mathcal{N} \subseteq \mathcal{L}_{\mathcal{N}}$ a set of norms, and $\mathsf{E} \subseteq \mathsf{Lit}$ the initial environment state. Scene S, with unique identifier s, is then defined as $\mathsf{S} = \langle s, \mathcal{R}, \mathcal{N}, \mathsf{E} \rangle$.*

Take a scene in an RPG that features the thief and a store owner in some store where goods can be purchased. In this scene it is most likely forbidden to take items without paying for them, or to damage the merchandise.[2] Now take a scene in which the thief and the store owner are joined in the store by a city guard. The same norms may apply as in the previous scene, but the thief now gives priority to ensuring nobody is around before stealing anything because of the presence of the guard, whereas the store owner might be more at ease in knowing that the eyes of the law are keeping watch over her belongings.

3.2 The Multi-Agent Game

A game is taken to be a composition of scenes. The way scenes are composed (the 'storyboard' of the game) is defined in a game specification, which identifies the scenes occurring in a game, and specifies when a specific scene makes a transition to another scene. Such transitions might depend on conditions being

[2] Note that the prima facie norms of $\mathcal{L}_{\mathcal{N}}$ do not allow for conditional statements, and it is therefore not possible to express statements such as the fact that it is obligatory to pay for an item after taking it.

fulfilled with respect to the environment of the scene, on specific actions being (jointly) executed by agents, or even some condition becoming true with respect to agents' mental states. To have a system of agents obey this specification, scene transition has to be operationalized in the semantics of the agent system. Because a detailed presentation of how the scene transition is realized does not contribute to the scope of the present approach, this is left unspecified and the game is taken to simply be a set of scenes, of the type defined in Def. 9.

Agents in 2APL [9] are defined by a configuration, which specifies their mental state in terms of goals, belief, plans, and rules. In this paper, we do not commit ourselves to an assumption about the specific language in which the agents are implemented, but do require that the behavior of agents is in accordance with the declarative specification of the role they enact, which contains elements that can be directly related to elements of agents' mental states, such as goals, information (beliefs) and behavioral rules. Specifically, the following assumptions and restrictions are enforced.

- *Every agent in the multi-agent game behaves in accordance with a role, such that the behavior of agents is completely described by the behavioral description which is part of the rules accompanying their role.*
- *The role of agents prescribes specific partially ordered goals, and the rules accompanying the role are taken to enable achievement of all these goals. However, it is not necessarily the case that every goal which the agent may have on grounds of its rules is part of the goals that the agent's role prescribes.*
- *It is assumed that agents do not interleave plans, even if they have multiple goals. If an agent has adopted multiple goals and has selected a plan based on the application of a behavioral rule for one of its goals, it will not apply a new rule until its selected plan is completed.*

The first scene of the multi-agent game is determined by the game's initial state, and consecutive scenes are determined as the game evolves; ie. as agents act in pursuit of their goals and 'things happen' in the game. Because it is not in the interest of the topic at hand, which is explanation and prediction of agents' behavior in the context of a multi-agent role-playing game, the operational transition of configurations of the multi-agent game will not be presented formally. Instead, it is assumed that the game takes place in some (known) scene, which provides a guideline with respect to behavior that can be expected of the agents populating the scene, as the behavior of each of them is based on some role.

4 Explaining and Predicting Agent Behavior

Mental state abduction can be used to abduce the mental state of BDI-based agents whose behavior can be observed. If this behavior is performed in the context of a multi-agent game, then information about the scene of the game and the role which agents enact helps improve the abduction process. If an agent's role is known, the set of rules the agent is taken to have at its disposition is reduced to the set of rules provided by the role, as the behavior descriptions in the rules of the roles completely describes behavior of the agents.

4.1 Explaining Agent Behavior

In the approach to mental state abduction as described in Sect. 2 (and in [5] in more detail), the behavior of an agent is explained on grounds of all rules this agent can be assumed to have if context is not considered. In the present setting, only the rules which are ascribed to the agent on account of its role in a particular scene are considered in the explanatory process. This ensures that the explanations provided for its behavior are *contextually grounded*, and that the set of rules which need to be considered is restricted in size.

The role of the agent contains behavioral rules and a partially ordered set of goals. There might exist agents which dutifully pursue the goals their role prescribes, and others which don't care about their role in the least. To capture these aspects of *role conformance*, two refined versions of the explanatory function are defined. Because the role-prescribed goals do not necessarily have a one-to-one correspondence with the goals that form the head of behavioral rules, a relation between the two has to be established. The functions g and b are defined, such that for $\omega = (\mathsf{goal}(\gamma) \wedge \mathsf{belief}(\beta))$, it holds that $g(\omega) = \{\gamma\}$ and $b(\omega) = \{\beta\}$. $\mathsf{Cn}(\Phi)$ denotes the closure of the set Φ under the consequence operator Cn, defined as $\mathsf{Cn}(\Phi) = \{\phi \mid \Phi \models \phi\}$.

Definition 10 (loosely role-conformant explanation) *Let $\delta \in \mathcal{L}_\Delta$ be a percept and $\langle r, (\Gamma, \leq_\Gamma), \mathcal{I}, \mathcal{BR}\rangle$ a role, as defined in Def. 8. The function $\mathsf{explain}_{lrc}$ for loosely role-conformant explanation is then defined as follows.*

$$\begin{aligned}
&\mathsf{explain}_{lrc}(\delta, \langle r, (\Gamma, \leq_\Gamma), \mathcal{I}, \mathcal{BR}\rangle) = \ (\Omega, \leq_\Omega) \\
&\quad \textit{where} \quad \Omega = \mathsf{explain}(\delta, {}^{\wp}\mathsf{desc}(\mathcal{BR})), \ \textit{and for any } \omega, \omega' \in \Omega \\
&\qquad\qquad \leq_\Omega = \ \{\ (\omega, \omega') \mid [\mathsf{Cn}(g(\omega)) \not\subseteq \mathsf{Cn}(\Gamma)] \wedge [\mathsf{Cn}(g(\omega')) \subseteq \mathsf{Cn}(\Gamma)]\ \} \\
&\qquad\qquad\quad \cup\ \{\ (\omega, \omega') \mid [\mathsf{Cn}(g(\omega)) \cap \mathsf{Cn}(\Gamma) = \emptyset] \wedge [\mathsf{Cn}(g(\omega')) \not\subseteq \mathsf{Cn}(\Gamma)]\ \} \\
&\qquad\qquad\quad \cup\ \{\ (\omega, \omega)\ \}
\end{aligned}$$

A rule can be said to be relevant to a role, if the goal for which this rule applies is in the closure of the role-derived goals. Thus, the rules for goals ϕ and ψ are both relevant to a role that prescribes the goal $\phi \wedge \psi$, just as the rule for $\phi \wedge \psi$ is relevant to a role that prescribes ϕ and ψ independently. The function $\mathsf{explain}_{lrc}$ maps to a poset of explanations, where the explanations are ordered on grounds of an ordering that ranks explanations containing role-derived goals over those with goals that derive from behavioral rules only. Explanations which contain exclusively role-derived goals rank over those with some role-derived goals, which in turn rank over explanations without role-derived goals.

Definition 11 (strictly role-conformant explanation) *The definition of the function $\mathsf{explain}_{src}$ is based on $\mathsf{explain}_{lrc}$, but takes into account the order on Γ.*

$$\begin{aligned}
&\mathsf{explain}_{src}(\delta, \langle r, (\Gamma, \leq_\Gamma), \mathcal{I}, \mathcal{BR}\rangle) = \ (\Omega, \leq_\Omega) \\
&\quad \textit{where} \quad \mathsf{explain}_{lrc}(\delta, \langle r, (\Gamma, \leq_\Gamma), \mathcal{I}, \mathcal{BR}\rangle) = (\Omega, \leq'_\Omega), \ \textit{and for any } \omega, \omega' \in \Omega \\
&\qquad\qquad \leq_\Omega = \ \{\ (\omega, \omega') \mid \exists \gamma \in \mathsf{Cn}(g(\omega)), \exists \gamma' \in \mathsf{Cn}(g(\omega')) : [\gamma <_\Gamma \gamma'] \wedge \\
&\qquad\qquad\qquad\qquad \neg\exists \gamma \in \mathsf{Cn}(g(\omega)), \exists \gamma' \in \mathsf{Cn}(g(\omega')) : [\gamma' <_\Gamma \gamma]\ \} \cup \ \leq'_\Omega
\end{aligned}$$

Strictly role-conformant agents are taken to also obey the priority ordering on goals specified by their role, and therefore $\mathsf{explain}_{src}$ takes this ordering into account this as well. Because not all goals need to be explicitly ordered, it is defined that some explanation ω is preferred to ω' on grounds of $\mathsf{explain}_{src}$ if and only if some goal γ, derived from ω, has explicit priority over some goal γ', derived from ω', and no goal derived from ω' has explicit priority over any goal derived from ω.

Instead of conforming to their role, agents might *rebel* against their role. Also, as explained in [12], agents which are allowed to have private objectives along with role-derived objectives can enact their roles in a selfish or social manner. This could imply an ordering which is the reverse of that seen in loose role conformance, or even of strict role conformance. Although it is not further dealt with, the fact that our approach allows for modeling explicit rebellion and different types of role enactment deserves pointing out.

4.2 Predicting Agent Behavior

An observed and explained sequence of actions can be regarded as the performed part of a trace. Given that the goal for which this plan was selected is still active, the agent can be expected to perform the remaining actions, which are the suffix of the trace of which the observed actions are the prefix. In explaining an agent's behavior, it was defined that a description of the agent's mental state can be regarded as an explanation. When predicting the behavior of the agent with respect to actions it has been observed to perform, multiple (distinct) action sequences may be predicted based on different assumed mental states. A predictive function is defined, taking these aspects into account.

Definition 12 (predictive function) *Let $\delta, \delta' \in \mathcal{L}_\Delta$ be percepts, $\omega \in \mathcal{L}_\Omega$ a mental state description and $\mathcal{RD} \in \mathcal{L}_{\mathcal{RD}}$ a set of rule descriptions. The function* $\mathsf{predict} : \mathcal{L}_\Delta \times \mathcal{L}_{\mathcal{RD}} \longrightarrow \wp(\mathcal{L}_\Omega \times \mathcal{L}_\Delta)$ *is then defined as follows.*

$$\mathsf{predict}(\delta, \mathcal{RD}) = \{\ (\omega, \delta') \mid (\omega, \delta\delta') \in {}^{\wp}\chi(\mathcal{RD})\ \}$$

Agents in a norm-governed society can be assumed to take norms into account in choosing their actions, either by design or by deliberation [15]. Similar to the explanatory functions taking into account role conformance of the agent (Defs. 10 & 11), one can consider *norm obedience* when predicting agent behavior. The norms of $\mathcal{L}_\mathcal{N}$ were defined to state about actions whether these are either forbidden (F) or obligatory (O). Informally, $F(\alpha)$ is taken to mean that the action $\alpha \in \mathsf{Act}$ is forbidden and that agents may be punished if they perform the action, whereas $O(\alpha)$ states that agents are obliged to perform action α and that may be punished if they do not perform it. Note that it is not defined what it means that the agent "*may* be punished", but the explanation that the behavior of the agent is somehow monitored (possibly by law-enforcing agents in the game), and that this monitoring is not infallible, should suffice.

Thus, it may occur that the agent performs a forbidden action, but gets away with it. The predicates forb and obl are defined on $\delta \in \mathcal{L}_\Delta$, such that

$$\begin{aligned} \mathcal{N} \models \mathsf{forb}(\delta) \quad & \textit{iff} \quad \exists \alpha, \delta', \delta'' \in \mathcal{L}_\Delta : [(\delta = \delta' \alpha \delta'') \wedge (F(\alpha) \in \mathcal{N})] \\ \mathcal{N} \models \mathsf{obl}(\delta) \quad & \textit{iff} \quad \exists \alpha, \delta', \delta'' \in \mathcal{L}_\Delta : [(\delta = \delta' \alpha \delta'') \wedge (O(\alpha) \in \mathcal{N})] \end{aligned}$$

Based on the above, a predictive function is defined which takes norm obedience into account. This function predicts a sequence of actions on grounds of an observed sequence of actions and behavioral rules, and relates it to the presumed mental state which would account for observed behavior if it were the agent's actual mental state. Moreover, this predictive function takes into account that norms may exist which forbid or oblige the agent to perform specific actions, as expressed in the ordering that ranks pairs with an action sequence containing some obliged but no forbidden actions above all others, and pairs with sequences that contain some forbidden but no obliged actions below all others.[3]

Definition 13 (norm-obedient prediction) *Let $\delta \in \mathcal{L}_\Delta$ be a percept, the tuple $\langle r, \Gamma_{\leq}, \mathcal{I}, \mathcal{BR} \rangle$ a role as defined in Def. 8 and $\langle s, \mathcal{R}, \mathcal{N}, \mathsf{E} \rangle$ a scene as defined in Def. 9. The predictive function $\mathsf{predict}_{no}$ is then defined as follows.*

$$\begin{aligned} &\mathsf{predict}_{no}(\delta, \langle r, \Gamma_{\leq}, \mathcal{I}, \mathcal{BR} \rangle, \langle s, \mathcal{R}, \mathcal{N}, \mathsf{E} \rangle) = \ (\Theta, \leq_\Theta) \\ &\quad \textit{where} \quad \Theta = \mathsf{predict}(\delta, {}^{\wp}\mathsf{desc}(\mathcal{BR})), \textit{ and for any } (\omega, \delta), (\omega', \delta') \in \Theta \\ &\qquad \leq_\Theta = \{\ ((\omega, \delta), (\omega', \delta')) \mid \mathcal{N} \models [\mathsf{obl}(\delta') \wedge \neg \mathsf{forb}(\delta')]\ \} \\ &\qquad\quad \cup \{\ ((\omega, \delta), (\omega', \delta')) \mid \mathcal{N} \models [\mathsf{forb}(\delta) \wedge \neg \mathsf{obl}(\delta)]\ \} \\ &\qquad\quad \cup \{\ ((\omega, \delta), (\omega, \delta))\ \} \end{aligned}$$

In Sect. 4.1 the remark was made that agents can explicitly rebel against their role. Similarly, agents might rebel against 'society', which can be modeled by means of explicitly presumed norm disobedience, such that traces with forbidden actions are considered to be preferred by the agent.

4.3 The Observer

To explain and predict behavior, an abstract external Observer is proposed (in line with our approach in [5]) which perceives the atomic observable actions performed by agents, attempting to explain those actions in context of the game and making predictions about actions it expects agents to perform next. The Observer maintains a model of each of the agents it observes, which contains the role the Observer attributes to the agent and a sequence of actions the agent has been observed to perform, along with explanations and predictions based on observed behavior in context of the attributed role.

Definition 14 (agent model) *Let R be a role of the type in Def. 8, $\delta \in \mathcal{L}_\Delta$ a list of perceived actions, $\Omega \subseteq \mathcal{L}_\Omega$ a set of explanations and $\Theta \subseteq \mathcal{L}_\Theta$. An agent model, with a unique identifier $i \in \mathsf{ID}$, is then defined as $\mathsf{A} = \langle i, \mathsf{R}, \delta, \Omega, \Theta \rangle$.*

[3] Note that a sequence with forbidden as well as obligatory actions is treated no differently than one that has only 'neutral' actions.

The Observer is assumed to have perfect observation of the environment and the actions agents perform. In many games, the roles of characters are evident from their external characteristics. The role might be indicated by the color of a suit, or simply by a label hovering over the character. In the following, it is assumed that agent i's role r can be deduced from the state of the environment, such that $\mathsf{E} \models \mathsf{enacts}(i, r)$. As scene transitions are taken to depend only on changes in the environment, the Observer always knows the scene in which the game takes place if it is made aware of the initial scene when the game starts, and is always correct about roles it attributes to agents.

Relaxing these assumptions — either by introducing more uncertainty on part of the Observer by design or because the game does not allow for perfect observation of the environment, scene transitions, or agents' roles — leads to interesting scenarios. Instead of just performing mental state abduction, the Observer is forced to perform role abduction and/or scene abduction as well. If observation of the environment or agents' actions is imperfect as well, yet more defeasibility is introduced. Given that our goal is to allow for designing agent-based game characters which have uncertainty about other characters' mental states, it is not in our interest to introduce any more uncertainty than necessary. Partial observation was discussed in [5], but here perfect observation is assumed.

Definition 15 (Observer) *Let $\mathcal{G}$ be a set of scenes as defined in Def. 9, and $s \in \mathsf{ID}$ a scene identifier such that $\langle s, \ldots \rangle \in \mathcal{G}$. The set of literals $\mathsf{E} \subseteq \mathsf{Lit}$ is the environment state and for every (perceived) agent i, A_i is an agent model. The Observer is then defined as $\langle \mathcal{G}, s, \mathsf{E}, \{\mathsf{A}_i, \ldots, \mathsf{A}_j\} \rangle$.*

The Observer as defined in this approach is an abstract entity, which serves to illustrate the explanatory and predictive process ultimately to be used by agents that observe other agents' behavior in some environment. For this reason the details of how the Observer configuration evolves with successive action observations and scene transitions are left to the imagination of the reader, and instead the focus is on the procedures defined in Sect. 4.1 and 4.2. Given a single sequence of observed actions for some agent i, the Observer can explain as well as predict this sequence of actions. Prop. 1 shows that each explanation — in terms of an agent's mental state — is accompanied by a matching prediction.

Proposition 1 (explanation matches prediction). *Given an agent model $\langle i, \langle r, \Gamma_<, \mathcal{I}, \mathcal{BR} \rangle, \delta, \Omega, \Theta \rangle$, where ${}^{\wp}\mathsf{desc}(\mathcal{BR}) = \mathcal{RD}$, $\mathsf{explain}(\delta, \mathcal{RD}) = \Omega$, and $\mathsf{predict}(\delta, \mathcal{RD}) = \Theta$, it holds that $\forall \omega \in \Omega : [\exists \theta \in \Theta, \delta' \in \mathcal{L}_\Delta : [\theta = (\omega, \delta')]]$.*

Proof. Def. 7 and Def. 12 show that explain *and* predict *are both based on ${}^{\wp}\chi$. In case of $\mathsf{explain}(\delta, \mathcal{RD}) = \Omega$, some $\omega \in \Omega$ iff $\exists (\omega, \delta'') \in {}^{\wp}\chi(\mathcal{RD}) : [\delta \preccurlyeq \delta'']$. For $\mathsf{predict}(\delta, \mathcal{RD}) = \Theta$, some $(\omega, \delta') \in \Theta$ iff $(\omega, \delta\delta') \in {}^{\wp}\chi(\mathcal{RD})$. It follows from Def. 6 that if $\delta \preccurlyeq \delta''$, then $\delta'' = \delta\delta'$ for some δ' (possibly $\delta' = \epsilon$), and therefore $(\omega, \delta') \in \Theta$. By definition of $\forall$, the proposition holds for $\Omega = \emptyset$.* □

Note that Prop. 1 extends to $\mathsf{explain}_{lrc}$, $\mathsf{explain}_{src}$ and $\mathsf{predict}_{no}$, as these are directly based on explain and predict. This is a very welcome fact, because it ensures that for every explanation a corresponding prediction can be made, also in

the case of the context-dependent explanatory and predictive functions. Based on role-conformant explanation some explanation may come out as 'top-ranked'. This can be considered the best explanation for the agent's behavior, and corresponding predicted behavior be regarded as the most probable. Prediction of ϵ — possibly indicating goal achievement — is outside of the scope of this approach, but very well worth further investigation.

It can occur that traces have overlapping segments. In such a case, the possibility exists that the Observer is able to explain a sequence of actions, but at a certain point observes an action that can neither be considered coherent with the currently presumed trace, nor as being the start of a new trace. This situation is visualized in Fig. 1, where δ' is the suffix of some trace $\delta\delta'$, as well as the prefix of another trace $\delta'\delta''$. Let α be the first action of δ'', and $\delta''' = \delta\delta'\alpha$. If the Observer explains $\delta\delta'$ as a coherent whole, then after perceiving α, it may be that $\mathsf{explain}(\delta''', \mathcal{RD}) = \emptyset$ because δ''' is not the prefix of any trace. It may, however, also be that α itself is not the prefix of any trace, such that $\mathsf{explain}(\alpha, \mathcal{RD}) = \emptyset$.

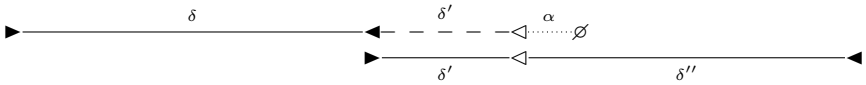

Fig. 1. Two traces, $\delta\delta'$ and $\delta'\delta''$, with an overlapping part δ'. The start of actual traces is denoted by ► and the end by ◄, explainable (segments of) potential traces end with ◁, and $\varnothing$ denotes a non-matching segment (ie. failure of explanation).

If the situation sketched in the previous paragraph occurs and explanation fails, then the Observer can backtrack along δ', starting at the end, until it finds a suffix $\delta'''' \succcurlyeq \delta'$ that can be explained in coherence with the last observed action α, such that $\mathsf{explain}(\delta''''\alpha, \mathcal{RD}) \neq \emptyset$. Given the assumption that agents are assumed to be able to completely execute their plans, the maximum overlap of any two traces of an agent's plans can be computed and used to give a measure of the maximum amount of backtracking the Observer has to perform. Let $\mathsf{len} : \mathcal{L}_\Delta \longrightarrow \mathbb{N}$ be a function that maps a percept to its length, such that $\mathsf{len}(\alpha_1, \ldots, \alpha_n) = n$, and let binds be predicate denoting that some sequence δ 'binds' the sequences δ' and δ'' together with overlapping action sequences, defined as

$$\mathsf{binds}(\delta, \delta', \delta'') \qquad \textit{iff} \qquad [(\delta \succcurlyeq \delta' \wedge \delta \preccurlyeq \delta'') \vee (\delta \succcurlyeq \delta'' \wedge \delta \preccurlyeq \delta')]$$

Given the definitions of len and binds, let $\mathsf{overlap} : \mathcal{L}_\Delta \times \mathcal{L}_\Delta \longrightarrow \mathbb{N}$ be a function that computes the overlap between two (distinct) action sequences, defined as

$$\mathsf{overlap}(\delta', \delta'') = \begin{cases} \mathsf{len}(\delta) \textit{ if } \exists \delta \in \mathcal{L}_\Delta : \mathsf{binds}(\delta, \delta', \delta''), \textit{ and } \delta' \neq \delta'', \textit{ and} \\ \qquad\qquad \neg\exists \delta''' \in \mathcal{L}_\Delta : [\mathsf{binds}(\delta''', \delta', \delta'') \wedge (\mathsf{len}(\delta''') > \mathsf{len}(\delta))] \\ 0 \quad \textit{otherwise} \end{cases}$$

Proposition 2 (backtrack with maximum trace overlap). *Given an agent model* $\langle i, \langle r, \Gamma_{\leq}, \mathcal{I}, \mathcal{BR} \rangle, \delta, \Omega, \Theta \rangle$*, where* $^{\wp}\mathsf{desc}(\mathcal{BR}) = \mathcal{RD}$ *and* $\delta = \delta' \alpha_1 \cdots \alpha_n$*, for which it is the case that* $\mathsf{explain}(\delta, \mathcal{RD}) = \emptyset$*,* $\mathsf{explain}(\delta' \alpha_1 \cdots \alpha_{n-1}, \mathcal{RD}) \neq \emptyset$*, and* δ' *is the complete trace of an actual plan executed by the agent, there exists a non-empty suffix* $\delta'' \succcurlyeq \delta$ *such that* $\mathsf{explain}(\delta'', \mathcal{RD}) \neq \emptyset$ *and* $\mathsf{len}(\delta'')$ *is smaller than or equal to one, plus the maximum overlap of any two traces of any plan which is part of the rules in* $\mathcal{BR}$*.*

Proof. Let $\Delta = \bigcup\{\, \tau(\pi) \mid (\gamma \leftarrow \beta \uparrow \pi) \in \mathcal{BR} \,\}$ *be the set of all (finite) observable traces of all plans part of the rules in the agent model. Because* δ' *is a complete plan trace,* $\mathsf{explain}(\delta', \mathcal{RD}) \neq \emptyset$ *and* $\delta' \in \Delta$*. Let* $\delta'' \succcurlyeq \delta$ *such that* δ'' *is the prefix of a trace of the agent's latest plan and it holds that* $\delta'' \succcurlyeq \alpha_1 \cdots \alpha_n$*. Then either* $\exists \delta''' \in \mathcal{L}_\Delta : [(\delta''' \preccurlyeq \alpha_1 \cdots \alpha_n) \wedge (\delta' \delta''' \in \Delta)]$ *such that the actual 'old' trace* δ' *is also the prefix of a misleading 'false' trace* $\delta' \delta'''$*, or not. If not, then* $\delta = \delta' \alpha$ *such that* $\alpha \succcurlyeq \delta$*,* $\mathsf{explain}(\alpha) \neq \emptyset$*, and* $\mathsf{len}(\alpha) = 1$*.*

If $\exists \delta''' \in \mathcal{L}_\Delta : [(\delta''' \preccurlyeq \alpha_1 \cdots \alpha_n) \wedge (\delta' \delta''' \in \Delta)]$*, then* δ''' *is the suffix of a 'false' trace* $\delta' \delta'''$ *and the strict prefix of* $\alpha_1 \cdots \alpha_n$*, such that* $\mathsf{len}(\delta''') < n$*. The 'new' trace, of which* δ'' *is the prefix, is started somewhere after the plan of which* δ' *is a complete trace has finished, such that* $\delta'' \succcurlyeq \delta' \alpha_1 \cdots \alpha_n$*. If the 'new' trace of which* δ'' *is the prefix is not started directly after* δ'*, because inbetween a complete trace of yet another plan was executed which together with* δ' *could be matched to a misleading trace, then* δ'' *is a strict suffix of* $\alpha_1 \cdots \alpha_n$*, such that* $\mathsf{len}(\delta'') > n$*. If* δ'' *is started directly after* δ'*, then the sequence* $\delta' \delta'''$ *and the sequence* $\delta'' = \alpha_1 \cdots \alpha_n$ *have an overlap of* $n - 1$*, which is the overlap of* $\delta' \delta'''$ *and the trace of which* δ'' *is the prefix, such that* $\mathsf{len}(\delta'') = n$*.*

Let x *be the maximum overlap of any two traces* $\delta_1, \delta_2 \in \Delta$*, such that* $(\mathsf{overlap}(\delta_1, \delta_2) = x) \wedge (\neg \exists \delta_3, \delta_4 \in \Delta : [(\mathsf{overlap}(\delta_3, \delta_4) = y) \wedge (y > x)])$*. Given that the trace* $\delta' \delta'''$ *and the 'new' trace of which* δ'' *is the prefix are both in* Δ *and have an overlap of* $n - 1$*, it holds that* $0 \leq n - 1 \leq x$*.* □

Prop. 2 can be guaranteed if the Observer does 'forget' any percepts and agents complete their plans. Especially the latter condition is unmaintainable in certain environments. If agents can drop their plans, 'freak' scenarios can arise. Take, for example, the case where $\alpha_1 \cdots \alpha_n$ is a plan trace, but the individual actions $\alpha_1, \ldots, \alpha_n$ are also the initial actions of individual plans. If an agent selects those plans in order, executing only the first action and then dropping the plan, the resulting sequence is indistinguishable from the trace.[4] However, because traces are finite and actions perfectly observable, Coroll. 1 still applies.

Corollary 1. *If agents drop their plans, then the Observer backtracks at most up to the length of the longest trace to find an explanation if* $\mathsf{explain}(\delta, \mathcal{RD}) = \emptyset$*.*

Proof. Let $\alpha_1 \cdots \alpha_n \in \Delta$ *be the longest trace. As* $\mathsf{explain}(\delta, \mathcal{RD}) = \emptyset$*, it must be that* $\exists \delta' \in \mathcal{L}_\Delta : [\delta' \succcurlyeq \delta]$ *and* δ' *is the prefix of the agent's current plan. In worst case,* $\delta = \delta'' \alpha_1 \cdots \alpha_n$ *for some* δ''*, such that* $\mathsf{explain}(\delta'' \alpha_1 \cdots \alpha_{n-1}) \neq \emptyset$*. After backtracking* $\mathsf{len}(\alpha_1 \cdots \alpha_n) = n$ *actions, Observer finds* $\mathsf{explain}(\alpha_1 \cdots \alpha_n) \neq \emptyset$*.* □

[4] One might ask whether explaining and predicting behavior has any benefit at all if such situations abound in some scenario, but that is not the point now.

5 Example

To illustrate the present approach, an example inspired by the popular role-playing game Oblivion [16] is introduced. Because of space limitations, some shorthand notation will be used. Lowercase predicate arguments represent ground atoms, and uppercase arguments represent variables. Our propositional language of course does not allow for variables, and therefore these are to be interpreted as a finite number of ground expressions, as should be clear from the context in which the notation is used. Spatial environments require moving around, and therefore $\mathsf{goto}(Loc)$ is defined, where the variable Loc stands for any valid location in the environment, and $(\phi?;\pi) + (\neg\phi?;\pi')$ means `if` ϕ `then` π `else` π'.

$$\mathsf{goto}(Loc) \equiv (\neg\mathsf{nearby}(Loc)?;\mathsf{walk\_towards}(Loc)) + (\mathsf{nearby}(Loc)?)$$

The scene S takes place in a store and features 'thief' and 'store owner' roles. The norm in this scene forbids stealing any item, as expressed in shorthand notation, such that $\mathsf{S} = \langle s, \{\mathsf{R}_t, \mathsf{R}_{so}\}, \{F(\mathsf{steal}(Item))\}, \mathsf{E}\rangle$. The 'thief' role prescribes the goal to have a particular item of interest ($\gamma = \mathsf{have}(item)$), and provides rules to achieve this goal. Also, a rule for exploring the store is provided.

$$\begin{aligned}
\mathsf{R}_t &= \langle thief, (\{\mathsf{have}(item)\}, \{(\gamma,\gamma)\}), \mathcal{I}, \{\mathsf{br}_{1_t}, \mathsf{br}_{2_t}, \mathsf{br}_{3_t}, \mathsf{br}_{4_t}\}\rangle \\
\mathsf{br}_{1_t} &= \mathsf{have}(item) \leftarrow \mathsf{distracted}(owner) \wedge \mathsf{in}(Cabinet, item) \uparrow \\
&\qquad \mathsf{goto}(Cabinet); \mathsf{open}(Cabinet); \mathsf{steal}(item); \mathsf{close}(Cabinet) \\
\mathsf{br}_{2_t} &= \mathsf{have}(item) \leftarrow \neg\mathsf{distracted}(owner) \uparrow \mathsf{goto}(owner); \mathsf{distract}(owner) \\
\mathsf{br}_{3_t} &= \mathsf{explored}(store) \leftarrow \neg\mathsf{explored}(cabinet_1) \wedge \ldots \wedge \neg\mathsf{explored}(cabinet_n) \uparrow \\
&\qquad \mathsf{goto}(cabinet_1); \mathsf{inspect}(cabinet_1); \ldots; \mathsf{goto}(cabinet_n); \mathsf{inspect}(cabinet_n) \\
\mathsf{br}_{4_t} &= \mathsf{ensured}(safety) \leftarrow \neg\mathsf{nearby}(Person) \uparrow \mathsf{double} - \mathsf{check_if_nearby}(Person)
\end{aligned}$$

The 'store owner' role R_{so} is left unspecified, except that it is stated she wants to protect her merchandise. In this paper procedural rules have not been discussed, but they may be allowed if only used for goal generation on grounds of events. If a customer breaks an object in the store, the perception of this fact prompts the store owner to adopt the goal to demand money from the culprit. This high-level approach remedies shortcomings in scripted character behavior in a natural way; in the game Oblivion it is possible, for example, to jump on the store counter or to smash objects without repercussions, because the store owner is only scripted to react to theft, and apparently not to vandalism.

The scene S transitions to some new scene S' upon entry of a city guard, as mentioned in Sect. 3.1, such that $\mathsf{S}' = \langle s', \{\mathsf{R}'_t, \mathsf{R}'_{so}, \mathsf{R}_{cg}\}, \{F(\mathsf{steal}(Item))\}, \mathsf{E}'\rangle$. The 'city guard' role R_{cg} is left unspecified, but it should suffice to say that the guard has merely come into the store to buy some item or chat with the store owner. If he becomes aware that someone is breaking the law (possibly the thief stealing the item), he may come into action and arrest the perpetrator. In the new scene the thief shows more cautious behavior because of the presence of the guard. This is illustrated by a change in the thief's role specification, such

that $\mathsf{R}'_t = \langle \mathit{thief}', (\{\gamma, \gamma'\}, \{(\gamma, \gamma), (\gamma', \gamma'), (\gamma, \gamma')\}), \mathcal{I}, \{\mathsf{br}_{1_t}, \mathsf{br}_{2_t}, \mathsf{br}_{3_t}, \mathsf{br}_{4_t}\}\rangle$. In this slighty changed role specification, it still is the case that $\gamma = \mathsf{have}(\mathit{item})$, but there is another role-prescribed goal $\gamma' = \mathsf{have}(\mathit{item}) \wedge \mathsf{ensured}(\mathit{safety})$ for which it is the case that $\gamma' >_\Gamma \gamma$.

Various possibilities exist for improvement, but lack of space forces us to skim over subtleties. More interesting is it to see how the Observer comes into play. Let $\mathcal{G}$ be the scenes of the game, and $\mathsf{A}_{\mathit{gent}} = \langle \mathit{gent}, \mathsf{R}_t, \mathsf{walk\_towards}(\mathit{cabinet}_1)\rangle$ the model of some agent called *gent*, such that the Observer has observed *gent*, in its role of *thief*, to perform the action of walking towards a certain cabinet. Let the Observer configuration for the first scene be $\langle \mathcal{G}, s, \{\mathsf{A}_{\mathit{gent}}\}\rangle$. Given the rules $\mathcal{BR}_t$ for the thief role, $\mathsf{explain}(\mathsf{walk\_towards}(\mathit{cabinet}_1), \mathcal{BR}_t)$ maps to a set of explanations $\Omega = \{\omega_1, \omega_2\}$, such that $g(\omega_1) = \{\mathsf{goal}(\mathsf{have}(\mathit{item}))\}$ and $g(\omega_2) = \{\mathsf{goal}(\mathsf{explored}(\mathit{store}))\}$. Based on role-conformant explanation, either loose or strict, $\omega_1 > \omega_2$ because having the item is a role-derived goal.

Prop. 1 states that every explanation is matched by a prediction. For ω_1, the tuples $(\omega_1, [\mathsf{open}(\mathit{Cabinet}), \mathsf{steal}(\mathit{item}), \mathsf{close}(\mathit{Cabinet})])$ (with the percept in Prolog-style list notation) are in the set of predictions. Given the small scenario and limited set of rules, this is the only possible prediction for the goal of having the item. If the thief is assumed not to be norm-obedient, the prediction that he will open the cabinet and steal the item comes out, which is plausible in this context. Assuming the thief actually is norm-obedient (which would be plausible in the scene S' where the guard is also present) gives a different picture. In that case another explanation for walking towards the cabinet can be considered best, if the corresponding predicted action sequence doesn't contain any forbidden actions. In this example only the goal to explore the store qualifies, but in a more extensive case this could include the goal to choose and purchase some item located in the cabinet.

The example in this section is inspired by an actual commercial role-playing game. It serves mainly to illustrate some of the focal points of the approach presented in this paper, and is necessarily limited in its scope and detail. Nevertheless, it should be sufficently rich to convince the reader of the fact that the high-level concepts of organizational modeling and agent programming apply transparently to the complex world of role-playing games. Moreover, the use of high-level social/intentional concepts has the additional benefit that these concepts can be reused for modeling, programming, and inter-character explanation and prediction of behavior.

6 Conclusion and Future Work

In this paper mental state abduction in the context of an agent-based game was described. A declarative game specification based on organizational principles such as roles, norms, and scenes, was introduced, and it was mentioned how it can be employed to have a system of autonomous agents behave in accordance with an intended storyline. An abstract Observer was said to observe the behavior of agents and provide explanations that take into account role-conformant

behavior, making the abduction process more efficient because it is based on a subset of rules, and ensuring that explanations are relevant to context. The Observer can also predict agents' future actions based on previously observed behavior, taking norm-obedience into account if the situation warrants this assumption. Role-conformant explanation and norm-obedient prediction have been shown to be complementary.

Future research should focus on explicitly taking models of the environment and agents' presumed mental states into account in the abductive process. Depending on the nature of the environment, it could be possible for an observer, be it an abstract entity or stiuated agent, to actively check whether specific (predicted) actions are possible, or whether an agent has achieved its goal or has some particular belief. Also, making use of norms in the explanatory process as well as the predictive process is to be considered. Finally, the path of formally investigating the multi-agent game as an operational system seems promising.

References

1. Loyall, A.B.: Believable Agents. PhD thesis, Carnegie Mellon University (1997)
2. Laird, J.E.: It knows what you're going to do: Adding anticipation to a Quakebot. In: AGENTS. (2001)
3. Albrecht, D.W., Zukerman, I., Nicholson, A.E.: Bayesian models for keyhole plan recognition in an adventure game. User Modeling & User-Adapted Interaction **8**(1-2) (1998) 5–47
4. Goultiaeva, A., Lespérance, Y.: Incremental plan recognition in an agent programming framework. Proceedings of PAIR (2007)
5. Sindlar, M.P., Dastani, M.M., Dignum, F., Meyer, J.-J.Ch.: Mental state abduction of BDI-based agents. In: Proceedings of DALT. (2008) 110–125
6. Scott, B.: Architecting a Game AI. In: AI Game Programming Wisdom. Charles River Media (2002) 285–289
7. Dignum, V.: A Model for Organizational Interaction. PhD thesis, SIKS Dissertation Series (2004)
8. Coutinho, L.R., Sichman, J.S., Boissier, O.: Modeling organization in MAS. In: SEAS. (2005)
9. Dastani, M.: 2APL: A practical agent programming language. Autonomous Agents and Multi-Agent Systems **16** (2008) 214–248
10. Pokahr, A., Braubach, L., Lamersdorf, W.: Jadex: A BDI reasoning engine. In Dastani, M., Dix, J., El Fallah Seghrouchni, A., eds.: Multi-Agent Programming. Springer (2005) 149–174
11. Tozour, P.: The Perils of AI Scripting. In: AI Game Programming Wisdom. Charles River Media (2002) 541–547
12. Dastani, M., Dignum, V., Dignum, F.: Role-assignment in open agent societies. In: Proceedings of AAMAS. (2003)
13. Dastani, M., Riemsdijk, M.B.V., Hulstijn, J., Meyer, J.-J.Ch.: Enacting and deacting roles in agent programming. In: Proceedings of AOSE. (2004) 189–204
14. Dignum, F.: Autonomous agents with norms. Artificial Intelligence and Law **7**(1) (1999) 69–79
15. Castelfranchi, C., Dignum, F., Jonker, C.M., Treur, J.: Deliberative normative agents: Principles and architecture. In: ATAL. (1999) 364–378
16. Bethesda Game Studios: The Elder Scrolls IV: Oblivion (2006)