Incremental generation of interaction moves for actors in a dynamic game environment

Johannes Graën
johannes.graen@studenten.ims.uni-stuttgart.de

Institut für maschinelle Sprachverarbeitung [1]
Universität Stuttgart
Pfaffenwaldring 5b
70569 Stuttgart

Diplomarbeit Nr. 111

Beginn	01.12.2011
Ende	31.05.2012
Betreuer	Prof. Dr. Leo Wanner
Prüfer	Prof. Dr. Hinrich Schütze

[1]Die Arbeit entstand in Kooperation mit der Gruppe TALN an der Universitat Pompeu Fabra (C/Tànger, 122–134, 08018 Barcelona, `http://taln.upf.edu/`).

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig verfaßt und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet.

Barcelona, den 31.5.2012	
Ort, Datum	Unterschrift

Acknowledgment

I wish to thank the following people (in alphabetical order) for their support from every point of view: Alicia Burga, Gabriela Ferraro, Graham Coleman, Simon Mille and Susana Bautista.

Abstract

Video games that make intensive use of dialogues – most notable role–playing games (RPGs) – tend to get around the issue of modeling natural language dialogues and rather keep to a pre–compiled list of turns from predetermined dialogue structures, of which the player merely has to select one. Dialogues with or among artificial agents (called non–player characters, NPCs) have hitherto been subject to various investigations, yet no efforts have been made so far to unite the generation of natural appearing dialogues with the generation of general interactions or actions at large.

In this work, I propose a model for describing games that comprise, apart from the world the game is set in, a dynamical list of actions (or moves) that a particular agent of the game world can perform at a certain point of time. The design of the game mechanism assures that both artificial and human players have the maximum freedom of choice at their disposal.

From the model, a concrete game setup is derived, consisting of both interactive and non–interactive types of actions that can be performed by the players of the game. The NPCs I provide for this particular scenario are provided with personality traits and base their decisions on inner reflections of the game world as well as on their characters and current states of mind.

Contents

1 Introduction

The preponderant majority of role–playing video game characters tends to appear artificial even after decades of video game development; see the introduction in (Wolf and Perron 2003). This diploma thesis' aim is to propose a model that facilitates the design of arbitrary games comprising a structure which allows the dynamic generation of natural interaction.

In what follows, I first provide an overview of *role–playing games* (RPGs), explain aspects of their user interfaces and exemplify the shortcomings of current examples.

Subsequently, I comment on other authors' thoughts about game design concerning the player's interaction with the game. A more profound view at the theory behind video games ("gaming theory", not to be confused with "game theory") cannot be given here, nonetheless, it is not needed to comprehend the challenges.

In the following two sections, I specify the motivation and the consequent objectives of this diploma thesis and, subsequently, illustrate its structure.

1.1 About role–playing games

Video games have accompanied the evolution of computers since the very first beginning of home computers.[1] Starting with simple text–mode–only programs, games have become more complex and graphically more sophisticated since the 1970s. Decreasing prices making home computers available for broad levels of the population as well as the development of home video game consoles made video games reach a wide audience. With increasing technical possibilities, the number of video game genres grew too, and nowadays, they have become as numerous as music or movie genres.

Among a wide range of video game genres, from *shoot–'em–ups* to *puzzle games*, there is one called *role–playing game* (RPG) originating from the correspondent non–electronic multiplayer *pen–and–paper game* that consists of several people, *players*, narrating parts of a fictional story in alternating order, each one being in charge of embodying one *(player) character* (PC).[2] All player characters are acting in an environment described and developed further by the *game master* (GM), who is also responsible for the actions of all *non–player characters* (NPCs), minor characters, who appear just once or once in a while, but do act in contrast to extras for reasons of a more realistic image. In some variants of this game, dice or other tools are used

[1] See (Wolf and Perron 2003) for an extensive overview.

[2] Uncommonly players do play more than a single PC, like, for example, in cases when a PC comes by another (temporary) PC as companion.

to bring randomness into the game stories by making the results of fights or the decisions of both PCs and NPCs dependent on the pips of the thrown dice, for instance.

When it comes to the video game adoption of pen–and–paper games, the earliest examples were text–only games with no proper NPC intelligence. Interaction with the world (as there were no individual NPCs) was achieved by entering simple, imperative text commands directed to the PC following a very rigid syntax.[3] These kinds of games would in today's terminology be called *text adventures*, if they are made for single–user play, and *multi–user dungeons* (MUD), if multiple players connect to a game server where they can interact with the game worlds as well as with the other players.

In the course of hardware progression, past limitations became obsolete and, among others, game world size, intelligence of the NPCs and, above all, game graphics advanced considerably. In times of graphical interfaces, pointing and clicking for the communication with the game replaced entering text commands in the majority of cases. The same applied to interaction with NPCs. If the player took the corresponding action to "speak" with an NPC, he was presented some text "uttered" by the NPC and a list of alternate answers for the PC, of which the player had to choose one. This kind of pre–compiled scripted dialogues is usually called *tree dialogues* (cf. (Bateman 2006, chapter 14; Wolf and Perron 2003, chapter 12)) because of its branching structure. Albeit called trees, tree dialogues can possess loops and are thus, technically seen, directed cyclic graphs in most cases. If no measures are taken within the game to break them (by, for instance, including loop counters into conditions for enabling/disabling edges), infinite loops are not prevented.

Regarding current single– and multi–player RPGs, it seems that game designers put all their efforts into developing naturally–looking game graphics and intelligent NPC behavior mechanisms. The PC–NPC dialogues remain scripted, repetitive and thus little realistic. In *massively multiplayer online role–playing games* (MMORPGs) many PCs populate together a huge 24/7 available game world. In these games, the communication most often takes place between individual players or within groups of players either in written or spoken form so that, although this kind of communication is one of the most essential parts of the game, it does not need to be modeled, as the game itself is not aware of the contents of these communications. On the other hand, for single–player RPGs the focus switches from player–player to PC-NPC communication.

[3] See the description of 'Zork' in (Brusk and Björk 2009) as an example.

Summing up, improvements concerning the communication interfaces of RPGs toward the use of less artificial, and thus more natural, ones would account for more realistic games and, in this way, catch up what game developers in the field of graphics already have achieved.

1.2 Game design

Many authors have addressed the question what the essence of "good" video games (including RPGs) is and how such games should be designed (cf. Frasca 2003; Gee 2007; Adams 2010). Regarding an RPG as a simulation of a particular world[4], the simulation becomes more alike to what it simulates the more freedom degrees it implements. In his article about "Interactive Storytelling", Crawford states:

> The first and foremost question in all interactive design is, "What does the user do? What kinds of choices can he make?" For interactive storytelling, the first–cut answer is simple: the user must be able to make dramatically interesting choices. Unfortunately, the universe of dramatic choices is stupendously large, and so rich and varied that no simple coding system exists that permits us to represent it manageably. We could never fit the universe of dramatic choices into a computer, nor can I imagine a conceptual scheme whereby such a universe could be expressed. (Crawford 2003, p. 262)

He argues in favor of leaving aside all non–dramatic options and thus comes to the conclusion that:

> [...] the player cannot use free–form input. In other words, the player cannot be allowed to type in any text (or speak anything into a microphone). The player must instead choose from a list of dramatic options we make available to him. This is nothing more than standard menu–driven input, which is easy to implement and process. (ibid., p. 263)

He then comes to develop a rudimentary model for a "dramatic universe" wherein 'verbs' (actions) can be taken by 'actors' on different 'stages' (locations) resulting in 'events' (ibid., pp. 264–267). This approach surely outperforms every however complex finite state automaton because the conditions

[4] "[...] video games are just a particular way of structuring simulation, just like narrative is a form of structuring representation." (Frasca 2003, p. 224)

for letting happen an 'event' are locally bound to the 'verb' and do not depend on an overall state of the whole game. Nonetheless, Crawford conceptually excludes the use of "free–form language" by reason of its complexity and gives priority rather to the overall "story" than to the player–game interface.

1.3 Motivation of the thesis

Dialogue trees and other fixed–menu techniques certainly do not come up to anyone's expectation of a realistic world simulation as they apparently narrow down the possibilities to act within the game world. A free–form input and output system and a free choice to do anything that is modeled by the game do correspond to today's seemingly unrestricted hardware capabilities. Furthermore, a natural simulation implies that different persons behave differently. In scripted dialogues, the author can use varying ways of speaking for every NPC he models, while for generic (generally template–based) dialogue definitions the same style applies to every NPC, which makes them seem indistinguishable.

An exemplary description of the impression, this kind of games leaves on the player, is given by Kerr and Szafron (2009):

> Imagine that you have just stepped into a tavern in a small town. You are here because this is where, according to the locals at least, you are most likely to find a little action. However, instead of a lively establishment full of patrons enjoying themselves, what you have found is more akin to what someone might expect to see in a wax museum. In place of lively patrons you find stiff unmoving people who all seem to repeat the same set of two or three phrases. (ibid., p. 154)

Summing up, a realistic RPG would, from my point of view, consist of:

1. a maximum freedom of choice, i.e. no "pruning" of less "dramatic" options as suggested by Crawford (2003),
2. "intelligent" NPCs, each of which possessing, apart from a particular view of the world, an own personality,
3. the NPCs' personalities manifesting themselves by means of natural language.

The free–form input of natural language texts brings up two challenges: On the one hand, players might tend to refer to out–of–world concepts, i.e.

things or actions that don't exist in the game world,[5] and hence the input needs repairing before it can be processed by the game. On the other hand, typing phrases takes more time than uttering the same phrase by speech. For this reason, RPGs tend to "freeze" the game time while a dialogue text is being entered, making dialogues a different game mode[6] and thus the game less realistic.

1.4 Objectives of the thesis

My aim is develop a model for games which:

1. describes the game world, the actions that can be executed in it and their effects on the world, the PCs and the NPCs (rather than describing storylines that players have to follow),

2. consists of an interface for any kind of artificial intelligence implementation for the NPCs residing in it,[7]

3. enables the generation of natural language texts by providing suitable output structures.

1.5 Structure of the thesis

Prior to defining my proper model in chapter 3, I give a summary of the state of the art, i.e. both video game approaches and investigative works on game interaction modelling in chapter 2. Chapter 4 describes a particular setup I use to demonstrate the abilities of the model defined in the preceding chapter, while chapter 5 goes into details of the implementation of the setup I carried out. Finally, the last chapter draws conclusions and outlines marks for further investigation.

[5] Also referred to as not being *diegetically consistent*. See, e.g., (Brusk and Björk 2009).

[6] Cf. the description of 'The Elder Scrolls III: Morrowind' and 'The Elder Scrolls IV: Oblivion' in (ibid.). In MMORPGs this won't work as many players are playing simultaneously. and changing the game mode would affect all off them.

[7] Whether the game world itself should be equipped with a kind of "intelligent" reactions to what happens in it remains undiscussed. The idea of underlying 'storytelling', i.e. foreseen proceedings, including concepts as 'foreshadowing' (giving hints about what possibly lies ahead) as an integral constituent of the player's game play experience has a lot to commend it.

1.6 Notations

The following notations will be used to refer to concepts introduced in chapter 3.

Variable	Description
Π	The game, a tuple containing all elements that constitute a game.
S	The stage, a description of the elements of the world the game plays in.
L	Alphabet of labels for all elements of the game.
R	Predicate definitions, determining the arity of all predicates.
W	The world state, a set of true propositions.
A	The set of all agents $a \in A$ in the game.
O	The set of all available operations.
T	The set of all triggers.
X_μ	The agent μ's obligations.
Q_μ	The agent μ's questions.
Δ	A triple consisting of operation, operator and operation parameters.
Θ	The set of all actors.
$\hat{A}_a$ $\hat{A}_h$	Sets of agents being played by an artificial / human actor.
$\mathcal{G}$	The game master, the authority that handles the gameplay.
φ	Game configuration, image of the variable parts of the game ($\langle W, A, \Theta, \mathcal{G} \rangle$) to a particular point in time.
Φ	The course of the game that can be derived by iteratively applying the function σ to the hitherto existing game configurations.

Constant	Description
$\varnothing$	Nothing or no value given.
?	Special case of $\varnothing$, when a truth variable is concerned.
$\top$	Tautology, used for formulae that are always true.
Ω	End symbol of a game.

Modification	**Description**
$W \leftarrow\Diamond_{p}^{\rho}$ $K_\mu \leftarrow\Diamond_{p}^{\rho}$ $B_\mu \leftarrow\Diamond_{p}^{\rho}$	Modification of the world state / the knowledge base of agent μ / the base of beliefs of agent μ setting the predicate p to the truthe value ρ
$X_\mu \leftarrow\bigcirc_{\Delta_2 \leftarrow \Delta_1}^{\nu \leftarrow \mu}$	Modification of the obligations of agent μ towards agent ν adding Δ_1 in return for Δ_2
$Q_\mu \leftarrow\Box_{p=\rho}^{\nu \leftarrow \mu}$	Modification of the questions of agent μ to agent ν with a predicate p and a truth value ρ.

Function	**Description**
$\dot{\rho}_W(p)$	Binary truth function for the world state.
$\dot{\rho}_{K_\mu}(p)$	Trinary truth function for the agent μ's knowledge base.
$\dot{\rho}_{B_\mu}(p)$	Multi–valued truth function for the agent μ's base of beliefs.
$f_\vartheta(a)$	Mapping function from agents to actors.
$g_a(\varphi_1, \ldots, \varphi_n)$	The game master's decision which agent should do the next move given the hitherto course of the game.
$g_\Delta(a, \vartheta)$	Decision of the actor ϑ which action the agent a played by himself wants to perform.
$g_g(\mathcal{G}, a, \Delta)$	Update of the game master $\mathcal{G}$ with the information that the action Δ should be executed in the name of the agent a.
$g_p(\Delta, W, A, \mathcal{G})$	Applies the execution of Δ to the game world. A modified world state and agent set are returned in case of success.
$g_i(\Theta, A, A')$	Update of the actors Θ with the modifications concerning their respective agents.
$\sigma(\varphi_1, \ldots, \varphi_n)$	Function to derive the game configuration for the next point in time given the previous configurations.

2 State of the art

In addition to extensive research in the field of dialogue at large, different approaches have been made to achieve natural interactions in game situations. Early works include (Stein and Gulla 1997), which focuses on "adaptive dialogue planning", i.e. awaiting the human counterpart's choice before making a decision, (Labrou and Finin 1998), which presents a language for agent–agent communication using definitions from (Searle 1969) called KQML and (Zukerman and Litman 2001), which gives an overview over both the recognition and the generation part of natural language including dialogue systems.

The following sections outline different categories of research that are related by some means or other. First, the so–called "dialogue games" stand for environments where typically two artificial agents 'play' together following the rules of the game. Those rules and the resulting gameplay are the essential. In the section about machine learning and planning, different concepts of artificial intelligence (AI) are applied to the problem of handling natural language, and in particular to dialogues. Other approaches try to extract information about dialogues in physical game environments from data collections of games played by human players. Finally, there are more video game specific works, which employ rule–based implementations aiming at improving the genuineness of the artificial characters by enhancing the behavior or language of it.

2.1 Dialogue games

In (Beun 2001), the term "Dialogue Game" stands for two artificial agents carrying out rule–based "communicative acts" by turns. Each of these acts affects the "cognitive state" of both agents, so that the basis for selecting an act is given by the preceding act of the other agent. Beun argues that dialogues are coherent, when the participants' "mental" configuration is in a consistent state, and that coherence can be achieved locally – in the context of the current acts –, when agents take their current cognitive state as basis of decision–making.[8] He highlights the benefit regarding the complexity of status–based decisions compared to approaches using planning.

Lebbink et al. tie their work up to Beun's definitions (see, among others, (Lebbink et al. 2003, 2004, 2005)). They advocate the use of *multi–valued logic* (MVL) in form of bilattices[9] in order to represent "inconsistent beliefs"

[8] This corresponds "uptaking" the "illocutionary force" in terms of Austin (1962).

[9] See (Ginsberg 1990) for the definition.

by propositions specifying the agent's view of the world; in all the mentioned dialogue games, agents follow the Gricean Maxims (cf. (Grice 1975)).

All the aforementioned works are restricted to fact–related communicative acts in dialogue situations (apart from asking and stating, their agents merely make use of acts on a meta–level like, for instance, concluding that they differ about a particular fact). Having no memory of actions preliminarily carried out, but, rather, only a list of intentions gained from prior acts, the obtained dialogues can result in quite unnaturally nested pairs of acts or even end in cycles, where the same respective question is posed as counter questions to the respective counterpart as Beun (2001) exemplifies.

2.2 Machine learning and planning

An early approach to apply the domain of artificial intelligence (AI) to video games is (Merrick and Maher 2006). The authors depict characters who evolve their behavior by means of learning algorithms. Rieser and Lemon also propose the use of machine learning algorithms to find the optimal "dialogue strategies" in (Rieser and Lemon 2008) and to decide which of the available information how to present to the counterpart in (Rieser and Lemon 2010). Both works are about helpful agents for real life assistance; the questions, however, concern universal questions regarding dialogues with artificial agents. Lemon (2011) claims to enhance the preceding works by combining the statistical models for content selection and language generation. Essentially along the same lines go Walker et al. (2007) and Cuayáhuitl (2009).

Koller et al. (2010) describe how planning can be used to generate interaction in terms of (Austin 1962). Koller and Petrick (2011) deal in detail with the benefit of applying planning to natural language generation commonly.

All the addressed works show how to make use of learning or planning algorithms to improve dialogues varying from handling spoken language to obtain intelligent agents. Most of them will complement my objectives by accounting for the implementation of the agents than compete with them.

2.3 Learning from game dialogues

Various approaches have been developed to derive criteria for the naturalness of dialogues, i.e. how natural dialogues should be, from collected data. Gorniak and Roy (2005) focus on linking speech to the physical (game) world, disambiguating statements with the additional information of the environment. For that purpose, they annotated one–directional spoken communicative situations, i.e. one player giving commands to the other one while the latter merely carrying out the commands remaining silent, and build a model

for understanding speech in given context (as it is the case in video games) on it.

Several later works made use of plenty of speech data from dialogue situations in a game. Reckman et al. (2010) try to extract phrases and their meaning from the physical context embedded data, while Orkin and Roy (2011) attempt to classify the data into speech acts as defined by Searle (1969) and in this vein attain a "model of social behavior". The source of their dialogue data is a publicly available game, where in each case two human players find themselves as a guest as well as a waiter in a restaurant situation with the developers' intention that they are supposed to apply their common knowledge about their respective roles in that context. The hereby gained statistical results serve (or shall serve in future works) as a base for better (in terms of naturalness) agent modeling in both cases.

Even closer to 'social' agents get McQuiggan and Lester (2006) with their "empirically informed models of empathy", nonetheless one of their human–driven agents is not allowed to use free speech, but merely to select of pre–compiled phrases.

The preceding works share one common aim, namely to enable artificial characters to act like human ones by equipping them with behavior extracted from data of human behavior in similar situations. Approaches like these could likewise integrate with my purposes.

2.4 Scripted dialogues

Brusk et al. (2007); Brusk (2008); Brusk and Björk (2009) treat the authenticity of dialogues with artificial game characters. The respective approaches make use of 'state chars' defined in (Harel 1987), which are an extension to finite state automata.

Mateas and Stern developed a language called ABL (A Behavior Language) in order to define "believable agents" among others for their video game 'Façade', further described in (Mateas and Stern 2003), which they categorize as "interactive drama'. ABL (explained in (Mateas and Stern 2004)) determines the actors' behavior in a form of event–based scripted stage directions. Several 'steps' can be executed sequentially as well as in parallel. They argue against actors' decision–making based merely on both the world and their inner state, because in their opinion this "principle of strong autonomy" contrasts with their claim that actors have to comply an overall storyline.

Kacmarcik (2005) specifies the requirements for question–answering for a RPG, while the author in (Kacmarcik 2006) works with recombinations of parse trees in order to create dialogue interactions. The latter work uses

searches within structures of characters' knowledge in the form of predicate logic entities in order to identify part that can be expressed by means of natural language.

Kerr and Szafron (2009) work with "intentional dialogue lines" which are neutral phrases that are mapped to phrases of a particular manner representing the speaker's character. As an example, they name the differentiation by axis of 'sophistication' and 'disposition' resulting in as many discrete cases (called 'bins' by the authors) as both dimensions multiplied with one another have (other dimensions could be 'race' or 'occupation'). The game designer is thought to specify as many cases as he wants to, while either he himself or any algorithm dynamically maps characters to cases.

The above mentioned works by Brusk et al. (2007); Brusk (2008); Brusk and Björk (2009); Mateas and Stern (2004) depict complex rule–based approaches to behavior design of artificial characters, yet all of them represent scripted actions including fixed concepts of activities, Kacmarcik (2005) merely focuses on the selection of knowledge entities for actions within a question–answering dialogue and Kerr and Szafron (2009) make a rather practical proposal of how to bring character dependent tones as representations of their personality into dialogue.

2.5 Résumé regarding this thesis

Summing up, most of the mentioned works are rather loosely related to the work proposed in this document, but can potentially be well incorporated representing aspects that are left out for not being in the main focus of this work.

In order to create a real video game under the requirements of artificial characters' naturalness of both language and behavior, many different approaches will need to be gotten colluding.

Some approaches come along with particular ideas that seem quite helpful to me, the description of cognitive states in (Beun 2001), the extension to classical binary truth values of (Lebbink et al. 2003, 2004, 2005) or the properties of ABL from (Mateas and Stern 2004), just to name a few.

3 The game model

The proposed model provides the ground onto which a specific game configuration can be set up and played. Allowing for an unrestricted set of possible configurations, the output depends heavily on the particular design.

The superficial structure of the model is quite plain: First of all, there is the so called *stage*, which includes everything that composes the physical world. Outside of the stage, there are (several) *actors* and one *game master*, whose function is similar to a game master in pen–and–paper games (see Figure 1).

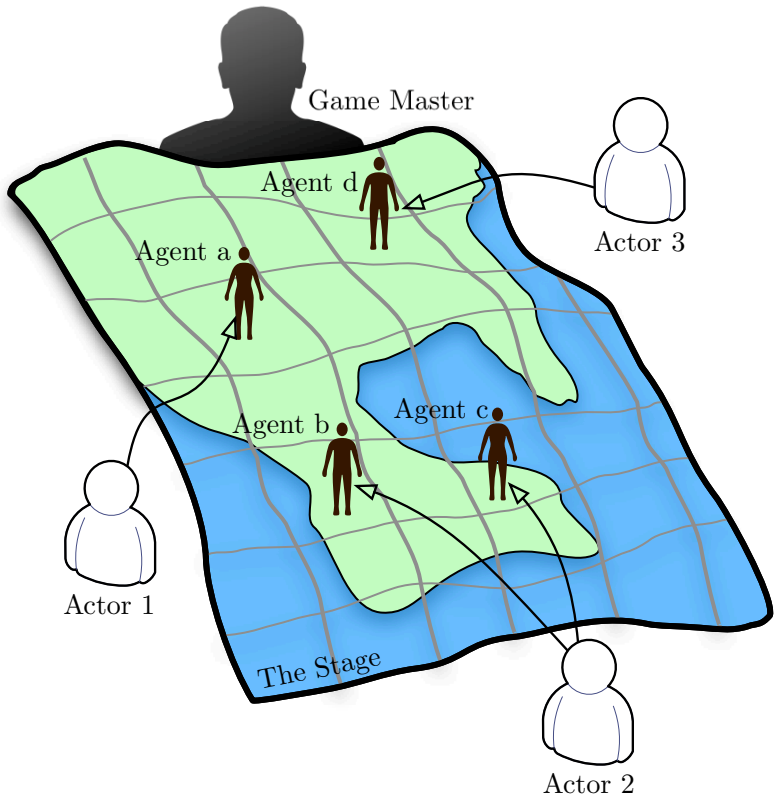

Figure 1: The stage being supervised by the game master. Three different actors 'play' four distinct agents.

Finally, the *gameplay* defines the circumstances under which these components collaborate so that a (hopefully dramatic[10]) story emerges. For this

[10]In terms of (Crawford 2003).

purpose, several functions are introduced to derivate, given the initial setup, the course of the game, a sequence of game *configurations* consisting of all variable parts of the game.

The game Π is thus defined as follows:

$$\Pi = \langle S, \Theta, \mathcal{G}, \sigma, f_\vartheta, h, g_a, g_g, g_\Delta, g_p, g_i \rangle \tag{1}$$

where

- S is the stage,
- Θ is a set of actors,
- $\mathcal{G}$ is the game master,
- σ is a function to derive a game configuration with the aid of the functions f_ϑ, h, g_a, g_g, g_Δ, g_p and g_i.

3.1 The stage

The *stage* S is a 6–tuple of the following kind:

$$S = \langle L, R, W, A, O, T \rangle \tag{2}$$

where

- L is an alphabet of labels further divided into distinct sets of element labels L_e, predicate labels L_r, truth value labels L_ρ, agent labels L_a, operation labels L_o and trigger labels L_t,
- R a set of pairs of a predicate label and a non–negative arity value,
- W a set of pairs called the world state,
- A a set of 5–tuples called agents,
- O a set of 4–tuples called operations and
- T a set of 4–tuples called triggers.

The set of truth value labels L_ρ contains at least the truth value labels ρ_1, ρ_0 and ρ_u, designating 'true', 'false' and 'unknown' respectively:

$$L_\rho \supseteq \{\rho_1, \rho_0, \rho_u\} \tag{3}$$

A *predicate definition* $r \in R$ is a pair of a predicate label l_r and a non–negative arity value n as stated above:

$$r = \langle l_r, n \rangle \text{ with } n \in \mathbb{N}^0 \tag{4}$$

For each predicate label l_r in L_r, there must be exactly one predicate definition containing l_r:

$$\forall l_r \in L_r \colon \exists \langle l_r, n \rangle \in R \wedge (\forall \langle l_r, n' \rangle \in R \rightarrow n = n') \tag{5}$$

The purpose of a predicate definition r is to define the number of arguments a *predicate* p possesses. A predicate p is made up of a predicate label l_r and a n–tuple of element labels $l_{\bar{e}}$ with n corresponding to the value linked with l_r in R:

$$p = \langle l_r, l_{\bar{e}} \rangle \text{ with } \langle l_r, n \rangle \in R \text{ and } \begin{cases} l_{\bar{e}} = \emptyset & \text{if } n = 0 \\ l_{\bar{e}} \in L_{e_1} \times \ldots \times L_{e_n} & \text{otherwise} \end{cases} \tag{6}$$

The *world state* W is a set of pairs w combining a predicate p with the truth value label ρ_1:

$$w = \langle p, \rho_1 \rangle \tag{7}$$

The set of all possible predicates is named P:

$$P = \bigcup p \tag{8}$$

Every potential predicate that is not contained by the world state set gets assigned the truth value ρ_0 by the world state truth function $\dot{\rho}_W$, other truth values are not permitted:

$$\dot{\rho}_W(p) = \begin{cases} \rho_1 \text{ if } \langle p, \rho_1 \rangle \in W \\ \rho_0 \text{ otherwise} \end{cases} \tag{9}$$

An *agent* $a \in A$ is a tuple of an agent label $l_a \in L_a$, a knowledge base K, a base of beliefs B, a set of obligations X and a set of questions Q:

$$a = \langle l_a, K, B, X, Q \rangle \tag{10}$$

The agent's *knowledge base* and *base of beliefs* are both sets of tuples of predicates and truth value labels, similar to the elements of the world state set W, whereas the knowledge base is allowed to contain the truth value labels ρ_1 and ρ_0 with an assigned default value of ρ_u for every predicate not

contained by any tuple of the set as defined in (11), while the base of beliefs can take every possible truth value label but ρ_u, also defaulting to ρ_u as defined in (12):

$$\dot{\rho}_K(p) = \begin{cases} \rho_x \text{ if } \langle p, \rho_x \rangle \in K \\ \rho_u \text{ otherwise} \end{cases} \tag{11}$$

$$\dot{\rho}_B(p) = \begin{cases} \rho_x \text{ if } \langle p, \rho_x \rangle \in B \\ \rho_u \text{ otherwise} \end{cases} \tag{12}$$

No predicate is allowed to occur in more than one tuple in both K and B:

$$\forall \langle p, \rho \rangle \in K \colon \forall \langle p, \rho' \rangle \in K \rightarrow \rho = \rho' \tag{13}$$

$$\forall \langle p, \rho \rangle \in B \colon \forall \langle p, \rho' \rangle \in B \rightarrow \rho = \rho' \tag{14}$$

Additionally, every element in K must be compatible to W, insofar as the world state truth function applied to any predicate p of a pair k in K must return the corresponding truth value label ρ from k:

$$\forall \langle p, \rho \rangle \in K \colon \rho = \dot{\rho}_W(p) \tag{15}$$

An *obligation* $x \in X$ is a 4–tuple of an agent label $l_{a'}$, representing the inquirer, an agent label $l_{a''}$, representing the addressee, an *action* Δ_1, represent the action promised – in return – by the inquirer, and Δ_2, representing the action required from the addressee, where Δ_1 can be omitted (i.e. be replaced by $\varnothing$):

$$x = \langle l_{a'}, l_{a''}, \Delta_1, \Delta_2 \rangle \tag{16}$$

Actions consists of an operation label l_o for the operation to be executed, an agent label l_a for the operator on behalf of which the operation should be executed and a tuple ψ made up of labels, predicates and (other) actions as parameters to the operation:

$$\Delta = \langle l_o, l_a, \psi \rangle \text{ with } \psi \in \left(L \cup P \cup \bigcup \Delta' \right)^n \text{ and } n \in \mathbb{N}^0 \tag{17}$$

As Δ' also denominates an action, the set of possible actions becomes infinite because of being defined recursively.[11] It depends on the particular setup to limit the set of possible actions to a reasonable set.

[11] Given that the particular setup defines an operation which consists of an action as parameter.

A *question* $q \in Q$ is a 4–tuple of an agent label $l_{a'}$, representing the inquirer, an agent label $l_{a''}$, representing the addressee, a predicate p and an omissible truth value label ρ:

$$q = \langle l_{a'}, l_{a''}, p, \rho \rangle \tag{18}$$

An *operation* $o \in O$ is a 4–tuple of an operation label l_o, an *event pattern* $\hat{e}$, a *precondition* $\breve{p}$ for the particular operation in the form of a logical formula and a set of *consequences* C:

$$o = \langle l_o, \hat{e}, \breve{p}, C \rangle \tag{19}$$

Triggers $t \in T$ are build alike with a trigger label l_t:

$$t = \langle l_t, \hat{e}, \breve{p}, C \rangle \tag{20}$$

An *event pattern* $\hat{e}$ is meant to match a particular operation that is executed (including operation label l_o, operating agent label l_a and a tuple ψ of labels, predicates and action as operation parameters) depicted by $\hat{e}_\Delta(l_o, l_a, \psi)$ with ψ corresponding to the definition in (17) or a modification made to:

- the world state (predicate p and truth value ρ) depicted by $\hat{e}_W(p, \rho)$,
- an agent's knowledge base (agent label l_a, predicate p and truth value ρ) depicted by $\hat{e}_{K_{l_a}}(p, \rho)$,
- base of beliefs (likewise) depicted by $\hat{e}_{B_{l_a}}(p, \rho)$,
- obligations (including agent labels l_a, $l_{a'}$ and $l_{a''}$, actions Δ_1 (optional) and Δ_2 being triples as defined in (17)) depicted by $\hat{e}_{X_{l_a}}(l_{a'}, l_{a''}, \Delta_1, \Delta_2)$ and accordingly $\hat{e}_{X_{l_a}}(l_{a'}, l_{a''}, \varnothing, \Delta_2)$ or
- questions (including agent labels l_a, $l_{a'}$ and $l_{a''}$, predicate p and an optional truth value label ρ) depicted by $\hat{e}_{Q_{l_a}}(l_{a'}, l_{a''}, p, \rho)$ and accordingly $\hat{e}_{Q_{l_a}}(l_{a'}, l_{a''}, p, \varnothing)$.

When matching an operation or a modification, the respective variables get bound to the particular parameters, so that any operation or trigger that makes use of this event pattern can get executed.[12]

An operation differs from a trigger in that operations are thought to be something that can be done on behalf of the agents in the game, while triggers shall represent side–effects like observations or emulations of inference

[12] They also serve for actors (see section 3.2) to survey changes to their agent's view of the game world.

rules. Therefore operation event patterns are intended in order to be used for operations, whereas all other types of event patterns remain for the use with triggers.[13]

A *consequence* is a tuple of a set of (unbound) variables V, a formula $\check{c}$ and a set of modifications M. There are different types of modifications of the modifiable parts of the game such as the world state W or the knowledge bases K, bases of beliefs B, obligations X or questions Q of any agent:

$$c = \langle V, \check{c}, M \rangle \tag{21}$$

In case of modifying the world state or the agent's knowledge or beliefs, a predicate and a truth value label must be provided. Obligation modifications consist, analogous to the definition of obligations in (16), of two agent labels and two operations – the first of them being optional – and a question modification consists, analogous to the definition of questions in (18), of two agent labels, a predicate and – optionally – a truth value label. The following handier alias definitions for modifications m and event patterns $\hat{e}$ will be used henceforth:

$$\begin{aligned}
m_W(p,\rho) &\triangleq W \leftarrow\!\Diamond_p^{\rho} & \hat{e}_W(p,\rho) &\triangleq W \widehat{\leftarrow\!\Diamond}_p^{\rho} \\
m_{K_{l_a}}(p,\rho) &\triangleq K_{l_a} \leftarrow\!\Diamond_p^{\rho} & \hat{e}_{K_{l_a}}(p,\rho) &\triangleq K_{l_a} \widehat{\leftarrow\!\Diamond}_p^{\rho} \\
m_{B_{l_a}}(p,\rho) &\triangleq B_{l_a} \leftarrow\!\Diamond_p^{\rho} & \hat{e}_{B_{l_a}}(p,\rho) &\triangleq B_{l_a} \widehat{\leftarrow\!\Diamond}_p^{\rho} \\
m_{X_{l_a}}(l_{a'}, l_{a''}, \Delta_1, \Delta_2) &\triangleq X_{l_a} \leftarrow\!\bigcirc_{\Delta_2 \leftarrow \Delta_1}^{l_{a''} \leftarrow l_{a'}} & \hat{e}_{X_{l_a}}(l_{a'}, l_{a''}, \Delta_x, \Delta_y) &\triangleq X_{l_a} \widehat{\leftarrow\!\bigcirc}_{\Delta_2 \leftarrow \Delta_1}^{l_{a''} \leftarrow l_{a'}} \\
m_{Q_{l_a}}(l_{a'}, l_{a''}, p, \rho) &\triangleq Q_{l_a} \leftarrow\!\Box_{p=\rho}^{l_{a''} \leftarrow l_{a'}} & \hat{e}_{Q_{l_a}}(l_{a'}, l_{a''}, p, \rho) &\triangleq Q_{l_a} \widehat{\leftarrow\!\Box}_{p=\rho}^{l_{a''} \leftarrow l_{a'}}
\end{aligned} \tag{22}$$

If a value is transferred from the world state W to the knowledge base K of an agent a, for instance, the element the truth value originates from replaces the truth value itself in the alias notation:

$$m_{K_{l_a}}(p, \dot{\rho}_W(p)) \triangleq K_a \leftarrow\!\Diamond_p^{W} \tag{23}$$

In case an optional element is omitted, the alias notation shows a question mark in place of it:

$$\begin{aligned}
m_{X_{l_a}}(l_{a'}, l_{a''}, \varnothing, \Delta_2) &\triangleq X_a \leftarrow\!\bigcirc_{\Delta_2 \leftarrow ?}^{l_{a''} \leftarrow l_{a'}} \\
m_{Q_{l_a}}(l_{a'}, l_{a''}, p, \varnothing) &\triangleq Q_{l_a} \leftarrow\!\Box_{p=?}^{l_{a'} \leftarrow l_{a''}}
\end{aligned} \tag{24}$$

Logical *formulae* consist of constants ($\top$, $\bot$), propositions (like $\dot{\rho}_W(p) = \rho$, $q \in Q_{l_a}$ or $x = y$), quantifiers ($\forall$, $\exists$) and connectives ($\neg$, $\wedge$, $\vee$, ...). Given

[13] Although there is technically no reason to limit their applications.

a set of instantiations for all (unbound) variables of a particular formula, the formula can be applied to said set as a filter to gain all instantiations satisfying it. The variables used by the formula must have been introduced by a corresponding event pattern or – as universal instantiations – by a variable list in the respective context.

3.2 Actors

Along with the game defined above, the "will" of the agents needs to become manifested for a game to unfold, otherwise simply nothing at all would happen in the game world. An actor $\vartheta \in \Theta$ is the decision–taking counterpart of an agent, like the mind for a body of an animate being[14]; the actor's only connection to the world (i.e. the stage) consists in his agent's experience. The relation between actors (Θ) and agents (A) is not further restricted than that there can be at most one actor per agent. On the contrary, one actor is allowed to "play" multiple agents[15] and there can be agents having no actor assigned to them (so they will not act nor 'react' at all).[16]

No requirements are defined for the internal structure of an actor as he can be played by human players too.[17] To keep the conventional terminology, agents played by human players are called *player characters* (PC) while the other ones are referred to as *non–player characters* (NPC).

A set of NPCs is referred to as $\hat{A}_a$, while a set of PCs is denominated with $\hat{A}_h$. Every a_i in (25) is an agent from the set A and any $\hat{a}_j$ refers to a mental representation for a particular agent. Since for human played agents there is no need to model said representation, a singleton agent in $\hat{A}_h$ corresponds to the pairs of agents and their mental representations in $\hat{A}_a$.

$$\begin{aligned} \hat{A}_a &= \{\langle a_i, \hat{a}_j \rangle, \ldots, \langle a_n, \hat{a}_m \rangle\} \\ \hat{A}_h &= \{\langle a_i \rangle, \ldots, \langle a_n \rangle\} \end{aligned} \tag{25}$$

[14]Since the actor does not come with any other way to access the world than by the experience of its agent (or agents), this combination cannot be compared to the top–view of board games, for instance, where each actor has full access to the whole board.

[15]This feature can come into play to realize game design concepts like 'foreshadowing' by coordinating the acting of groups of agents.

[16]Dynamical changes of actor–agent assignments can also be imagined such as a 'demon' taking control over an agent.

[17]And I do not want nor am I able to define internal structures for humans.

Each actor $\vartheta \in \Theta$ is then defined according to whether it concerns a human or an artificial player. Artificial players possess an additional value ζ to keep track of their plans, for instance:

$$\vartheta = \begin{cases} \langle \hat{A}_a, \zeta \rangle & \text{if the actor is artificial} \\ \langle \hat{A}_h \rangle & \text{if the actor is human} \end{cases} \tag{26}$$

The function f_ϑ maps an agent to the corresponding actor if there is any. No agent can be part of more than one tuple of either $\hat{A}_a$ or $\hat{A}_h$ and each actor plays at least one agent, thus making f_ϑ a surjective mapping of agents to actors:

$$f_\vartheta(a) = \begin{cases} \vartheta & \text{if a is assigned to an actor} \\ \varnothing & \text{otherwise} \end{cases} \tag{27}$$

Actors are granted only a restricted sight at the stage. They know the 'rules of the world', i.e. the part of the stage that consists of labels, predicates, operations and triggers, but cannot access any agent's values but their assigned agents' ones. Direct access to the world state is equally not permitted to any actor. New information can only be gained from the assigned agents' values.

Artificial actors may be built up in different ways, ranging from quite simple implementations which decide on what they "want" to do by chance to "intelligent" implementations keeping track of its agent's values and continuously working out plans based on these records. In cases of goal directed actor implementations, these goals must either be incorporated into the actor directly or any kind of motivation based concept must be made available to him to generate them dynamically.

One suitable model of personality an actor can have at its disposal would simply map the factors of the so called 'Big Five model of personality'[18] and thus serve for evaluating individual values for certain actions.

3.3 The game master

The particular instantiation of the game model defines what the game is about and which actions can be taken on behalf of which agents and under which circumstances. Parallel execution of two or more operations is not intended as this would increase the complexity of operation definitions while

[18]For diverse aspects of this versatile model see (Costa and McCrae 1992; McCrae and Costa 1999, 2003; John et al. 2008; Srivastava 2010; Mairesse and Walker 2010).

providing little benefits, since for human capabilities interactions must rather be sequential than parallel.

It is up to the *game master* (GM) referred to by $\mathcal{G}$ to decide who (i.e. which actor) to ask for performing next. Criteria for its decision may be, among others:

- which actor had most time to wait for getting the turn,
- which actor shows most 'urgency' to act (due to an inner state of the actor or the current game situation for example),
- which actor does the game master prefer to react to the current game situation[19].

When being told the chosen actor's decision, the game master is in charge to verify the conditions necessary for that agent to carry out the chosen operation, apply the consequences to the stage and take care of triggers that might get fired. After no trigger is firing anymore, the game master passes a list of all relevant modifications (i.e. every modification concerning the particular agent's knowledge base, base of beliefs, obligations or questions) to every actor.

3.4 Gameplay

The gameplay is the temporal course of a game; it defines how the game starts and ends as well as how the constituents mesh together during the game. As an aggregation of all rules of the game, it resembles the manual of a classical board game.

Based on a 4–tuple φ called configuration which consists of all variable parts of the game, namely the world state W and the agent set A of the stage S as well as the set of actors Θ and the game master $\mathcal{G}$, the complete course Φ can be represented as a sequence of n configurations, each of which possessing the variable parts for a particular point in time. The function σ returns a configuration φ_i based on the preceding configurations φ_0 to φ_{i-1} for the subsequent point in time[20]. In order to access parts of a tuple, the notation $h_i(t)$ is used, where t is the tuple to be accessed and i the particular index:

$$\Phi = (\varphi_0, \ldots, \varphi_n) \tag{28}$$

[19]The game master could as well feature goals and intentions in order to obtain a more 'dramatic' story.

[20]Time is to be seen as as sequence of discrete steps. Continuous time involves the problem of parallelism.

so that

$$\varphi_0 = \langle h_W(S), h_A(S), \Theta, \mathcal{G} \rangle \tag{29}$$

and

$$\varphi_i = \sigma(\varphi_0, \ldots, \varphi_{i-1}) \tag{30}$$

with

$$\forall j < n \colon \sigma(\varphi_0, \ldots, \varphi_j) \neq \Omega \tag{31}$$

and

$$\sigma(\varphi_0, \ldots, \varphi_n) = \Omega \tag{32}$$

The function σ either returns the subsequent configuration or Ω, if the end condition of the game master is fulfilled:

$$\sigma(\varphi_0, \ldots, \varphi_m) = \begin{cases} \Omega & \text{if } a = \varnothing \\ \langle h_W(\varkappa), h_A(\varkappa), \Theta, h_{\mathcal{G}}(\varkappa) \rangle & \text{otherwise} \end{cases} \tag{33}$$

with

$$a = g_a(\varphi_0, \ldots, \varphi_m) \tag{34}$$

and

$$\varkappa = g_p(\Delta, h_W(\varphi_m), h_A(\varphi_m), g_g(h_{\mathcal{G}}(\varphi_m), a, \Delta)) \tag{35}$$

$$\Theta = g_i(h_\Theta(\varphi_m), h_A(\varphi_m), h_A(\varkappa)) \tag{36}$$

$$\Delta = g_\Delta(a, f_\vartheta(a)) \tag{37}$$

The function g_a represents the game master's decision about which agent should make a move given all previous versions of the world state, the actors and the game master itself. At first, the game master checks whether the end condition are complied with and returns $\varnothing$ in this case:

$$g_a(\varphi_0, \ldots, \varphi_m) = \begin{cases} \varnothing & \text{if end conditions are met} \\ a & \text{otherwise} \end{cases} \tag{38}$$

Subsequent to the determination of the agent to do the next move, the corresponding actor is asked what he wants the determined agent to do. If there is no profitable option to act in terms of the internal evaluation of the particular actor, the function returns $\varnothing$ instead of an action Δ:

$$g_\Delta(a, \vartheta) = \begin{cases} \varnothing & \text{if the actor decides to do nothing} \\ \Delta & \text{otherwise} \end{cases} \tag{39}$$

Given the chosen agent a and action Δ, an update of the game master itself is returned by the function g_g:

$$g_g(\mathcal{G}, a, \Delta) = \mathcal{G}' \tag{40}$$

Applying the action Δ to the stage results in both a possibly modified world state and possibly modified agents. In case the chosen action is not appliable, the function g_p returns $\varnothing$, otherwise a pair of modified world state and agents. The game master needs to know, if the application fails, in order to account for that when indicated:

$$g_p(\Delta, W, A, \mathcal{G}) = \begin{cases} \varnothing & \text{if the action cannot be executed} \\ \langle W', A', \mathcal{G}' \rangle & \text{otherwise} \end{cases} \tag{41}$$

To finalize one step, the actors experience the changes of their respective agents:

$$g_i(\Theta, A, A') = \Theta' \tag{42}$$

The most minimalist game consists of the standard truth value labels, at least one predicate label and accordingly one predicate, one agent label as well as a corresponding agent and one operation label together with an operation being based upon it. When the game starts (and it is supposed to be terminable), the only existing agent is allowed to execute the only operation which toggles the truth value of the only predicate (featuring no argument) and thus meets the conditions for the game to end (defined like this beforehand). The course of this game is depicted by figure 2.

For other more complex games control is given to the game master, when the stage and the actors are at their initial configurations. The game master selects an actor to choose an action (or to forego) and applies the consequences of the execution to the stage (if needed). The results in terms of modifications concerning the operating agent or a failure[21] are returned thereupon to the actor. The actor can – if applicable – draw conclusions from a failure and extend his agent's knowledge base or base of beliefs accordingly. If he is allowed another try by the game master, his decision would at best be different this next time.

After every turn that the game master managed, game rules are applied in order to check if end conditions meet or not (like the 'death' of all agents for instance, or the human controlled agent achieving his goal). End conditions

[21] I.e. the game master refusing the execution of the chosen action due to unmet world state conditions, which were not present at the agent's knowledge base at the time the corresponding actor took his decision.

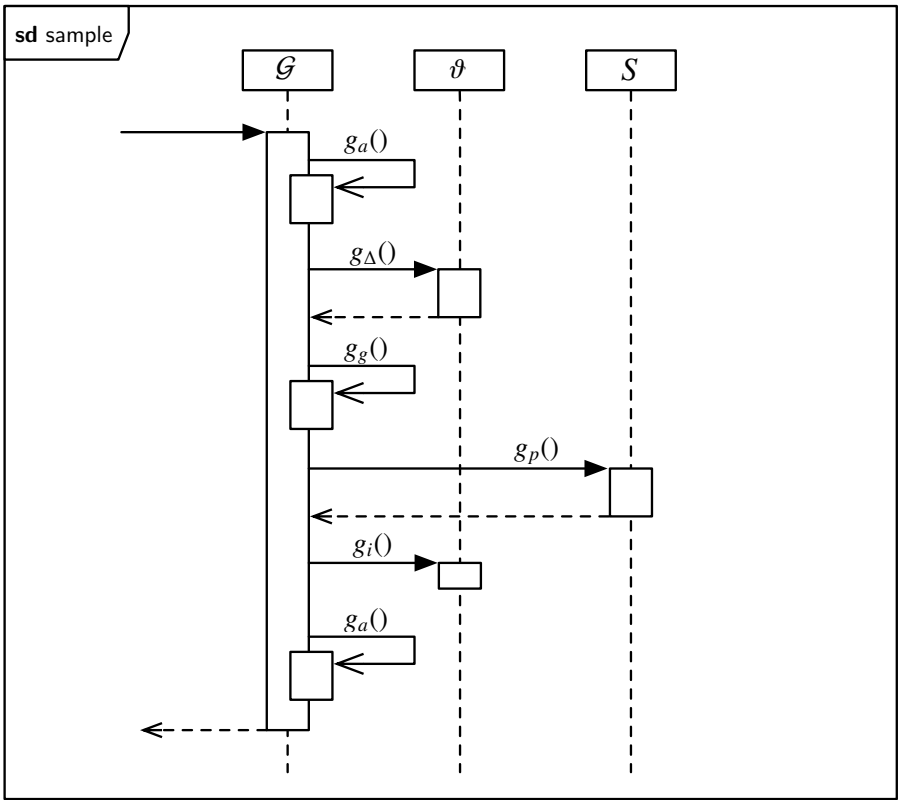

Figure 2: Exemplary course of a minimalist game featuring the game master $\mathcal{G}$, an actor ϑ and the stage S.

are up to the particular game setup and could be none (which would result in an infinite game). Gameplay rules may also set time limits for actor decisions, so that not managing to come to a decision in time causes the actor automatically either to choose a default option or to forego.

4 Particular setup of a game

My setup focuses on aspects of interaction between agents. Accordingly, the majority of the operations that can be executed on behalf of one agent addresses another agent.

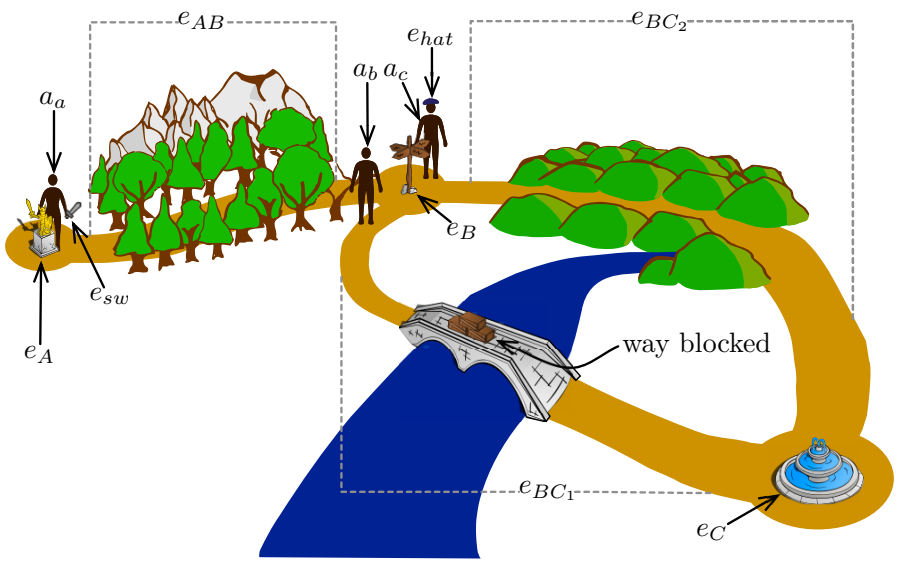

Figure 3: Pictorial representation of the physical part of the game world described in chapter 4.

The scenario, also referred to as *game world*, comprises a number of characters in a scenery consisting of places that are connected by roads as illustrated by Figure 3. These characters, hereafter solely referred to as *agents*, have limited knowledge about the world they are in. They can broaden it by exploring the world and asking questions among each other.

Every agent comes with preferences in the form of goals which he tries to accomplish in the course of the game. There may be situations where one agent needs another agent to cooperate with him for his desires to become reality.

In what follows, the overall descriptiveness of the setup in terms of labels and predicates will be specified in sections 4.1 and 4.2. The initial world state will be defined in section 4.3 as well the initial knowledge of the agents about the world in 4.4. In the subsequent sections, 4.5 and 4.6, the set of available operations and the thereby activatable triggers are defined.

To complete the game, actor are set up with mental configurations in section 4.7, and the game master and the gameplay are defined in 4.8. Finally,

commented game traces are given for the stated setup as well as for slight alterations of it in section 4.9.

4.1 Labels and their meaning

The following labels are used in the setup:

$$L_e = \left\{e_A, e_B, e_C, e_{AB}, e_{BC_1}, e_{BC_2}, e_{sw}, e_{hat}, e_{ring}\right\} \tag{43}$$

$$L_r = \left\{r_{loc}, r_{ag}, r_{pw}, r_{th}, r_{blk}, r_{wa}, r_{pos}, r_{has}, r_{cub}, r_{po}, r_{ws}\right\} \tag{44}$$

$$L_\rho = \left\{\rho_1, \rho_0, \rho_u\right\} \tag{45}$$

$$L_a = \left\{a_a, a_b, a_c\right\} \tag{46}$$

$$L_o = \left\{l_{use}, l_{unbl}, l_{give}, l_{comm}, l_{prom}, l_{quest}, l_{assert}\right\} \tag{47}$$

$$L_t = \left\{l_1, l_2, l_3, l_4, l_5, l_6, l_7\right\} \tag{48}$$

The labels e_A, e_B and e_C designate locations, e_{AB}, e_{BC_1} and e_{BC_2} pathways between these locations and e_{sw}, e_{hat} as well as e_{ring} a sword, a hat and a ring, respectively[22].

The predicates denominated by the labels r_{loc}, r_{ag}, r_{pw} and r_{th} define their arguments as being a location, an agent, a pathway and a thing, respectively. A pathway connects by definition exactly two different locations. This fact is specified by the predicate denominated by the label r_{po}, which states that the given location is a point of the given pathway. The label r_{blk} applied to a pathway defines whether that pathway is blocked or not; r_{wa} applied to a thing represents the wearability of it, while r_{ws} serves to define that the given agent is wearing the given thing. The predicate labels r_{pos}, r_{has} and r_{cub} express the states of an agent to be at a certain location, to own a certain thing and to be able to unblock a certain pathway, respectively.

As for the truth values, my setup settles for the minimum of ρ_1, ρ_0 and ρ_u. Finer grained truth values are not needed, since the belief of an agent will not be modeled in this setup.

The three agents addressed to by the agent labels in L_a are further specified in section 4.4.

For every operation label contained in L_o, an operation is defined as elaborated in section 4.5.

The triggers I defined in section 4.6 fire as reaction to changes to either the (physical) world state or the knowledge base of an agent. The former

[22]Since all entities in my setup are unique, i.e. nothing countable exists, the use of the definite article e.g. for 'the ring' is justified.

usually involves changes to some agent's knowledge base[23], which can be subsumed under 'observations', while the latter only changes the same agent's knowledge base or base of beliefs, what can be taken as 'reasoning', although he will not arrive to any conclusion in this way, because drawing conclusions belongs to the related actor. In my setup, the latter type of triggers is used exclusively for logic constraints like that an agent can only be at one place at once. Since triggers do not represent real world concepts in either case, the assigned labels are just enumerated.

4.2 Predicates

As stated above, the predicates listed in L_r take either one or two arguments:

$$R = \Big\{\langle r_{loc}, 1\rangle, \langle r_{ag}, 1\rangle, \langle r_{pw}, 1\rangle, \langle r_{th}, 1\rangle, \langle r_{blk}, 1\rangle, \langle r_{wa}, 1\rangle, \\ \langle r_{pos}, 2\rangle, \langle r_{has}, 2\rangle, \langle r_{cub}, 2\rangle, \langle r_{po}, 2\rangle, \langle r_{ws}, 2\rangle\Big\} \tag{49}$$

For any predicate in (49) with an argument count of two, the first one is always meant to be an agent label.

4.3 The world state

The world state is a complete set of all true propositions that describe the current state of the game world. The world state setup in (50) reads as follows: a_a, a_b and a_c are agents, e_A, e_B and e_C locations, e_{AB}, e_{BC_1} and e_{BC_2} pathways connecting e_A and e_B (the former) as well as e_B and e_C (the latter two), e_{sw}, e_{hat} and e_{ring} are wearable things owned by a_a, a_b and a_c, respectively, where e_{sw} and e_{hat} are actually worn by their owner. a_a is at location e_A, while a_b and a_c are at e_B. The pathway e_{BC_1} is blocked and only a_b is able to unblock it.

$$W = \Big\{\Big\langle\langle r_{loc}, \langle e_A\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{loc}, \langle e_B\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{loc}, \langle e_C\rangle\rangle, \rho_1\Big\rangle, \\ \Big\langle\langle r_{pw}, \langle e_{AB}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{pw}, \langle e_{BC_1}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{pw}, \langle e_{BC_2}\rangle\rangle, \rho_1\Big\rangle, \\ \Big\langle\langle r_{th}, \langle e_{sw}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{th}, \langle e_{hat}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{th}, \langle e_{ring}\rangle\rangle, \rho_1\Big\rangle, \\ \Big\langle\langle r_{wa}, \langle e_{sw}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{wa}, \langle e_{hat}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{wa}, \langle e_{ring}\rangle\rangle, \rho_1\Big\rangle, \\ \Big\langle\langle r_{ag}, \langle a_a\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{ag}, \langle a_b\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{ag}, \langle a_c\rangle\rangle, \rho_1\Big\rangle, \tag{50}$$

[23]Consequences changing the proper world state are to be interpreted as (physical) reactions in the world.

$$\Big\langle\langle r_{po}, \langle e_{AB}, e_A\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{AB}, e_B\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{BC_1}, e_B\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{po}, \langle e_{BC_1}, e_C\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{BC_2}, e_B\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{BC_2}, e_C\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{pos}, \langle a_a, e_A\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{pos}, \langle a_b, e_B\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{pos}, \langle a_c, e_B\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{has}, \langle a_a, e_{sw}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{has}, \langle a_b, e_{ring}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{has}, \langle a_c, e_{hat}\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{ws}, \langle a_a, e_{sw}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{ws}, \langle a_c, e_{hat}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{blk}, \langle e_{BC_1}\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{cub}, \langle a_b, e_{BC_1}\rangle\rangle, \rho_1\Big\rangle\Big\}$$

Changing the world state means adding new proposition or removing existing ones from the set when they are thought to become true or false, respectively.

4.4 Agents

Every agent's knowledge can be seen as the relative complement of what the agent does not know of the state of the world. As the world state consists only of true propositions, adopting it for the agent's knowledge means – in terms of sets – to build the union of the world state as it is with the predicates of its complement combined to negative propositions. In the definitions (51), (52) and (53), I indicate the relative complement in the world of all true propositions not known by the particular agent in the world state, unified with the set of false propositions that the agent does know.

The (by definition female) agent referred to by a_a does not know anything about the other two agents, neither does she know that the pathway e_{BC_1} is blocked or about the existence of the alternative pathway e_{BC_2}, the hat or the ring. She does know from her initial setup that e_C is not part of the pathway connected to e_A[24], that she is nowhere else than at e_A[25], while no one else but her is there.

$$K_{a_a} = W \setminus \Big\{\Big\langle\langle r_{ag}, \langle a_b\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{ag}, \langle a_c\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{blk}, \langle e_{BC_1}\rangle\rangle, \rho_1\Big\rangle, \tag{51}$$
$$\Big\langle\langle r_{unbl}, \langle a_b, e_{BC_1}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{has}, \langle a_b, e_{ring}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{has}, \langle a_c, e_{hat}\rangle\rangle, \rho_1\Big\rangle,$$
$$\Big\langle\langle r_{pw}, \langle e_{BC_2}\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{BC_1}, e_B\rangle\rangle, \rho_1\Big\rangle, \Big\langle\langle r_{po}, \langle e_{BC_1}, e_C\rangle\rangle, \rho_1\Big\rangle,$$

[24]The knowledge, that every pathway always has two points connected to it and the ability to deduce that knowing two points of a particular pathway makes all other points not part of it, are encoded in operation and trigger definitions. Initially, these facts must explicitly be given, since 'reasoning' happens only as a reaction to the execution of an operation and concerns only this operation's circumstances.

[25]This is another elementary fact which has to be heeded when specifying the initial setup.

$$\Big\langle\langle r_{pos},\langle a_b,e_B\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{pos},\langle a_c,e_B\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{th},\langle e_{hat}\rangle\rangle,\rho_1\Big\rangle,$$
$$\Big\langle\langle r_{th},\langle e_{ring}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{wa},\langle e_{hat}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{wa},\langle e_{ring}\rangle\rangle,\rho_1\Big\rangle,$$
$$\Big\langle\langle r_{ws},\langle a_c,e_{hat}\rangle\rangle,\rho_1\Big\rangle\Big\}\cup\Big\{\Big\langle\langle r_{po},\langle e_{AB},e_C\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_a,e_B\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_a,e_B\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_a,e_C\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_b,e_A\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_c,e_A\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{has},\langle a_b,e_{sw}\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{has},\langle a_c,e_{sw}\rangle\rangle,\rho_0\Big\rangle\Big\}$$

The agents referred to by a_b and a_c have no knowledge about the existence of a_a and (therefore) do not know where she is or that she owns or wears the sword, nor that the sword even exists (similarly, a_c does not know about a_b's ring). Both agents share the knowledge about the pathway e_{BC_1} being and the pathways e_{AB} and e_{BC_2} not being blocked, about themselves not being at any other location than in e_B and no other agent being there with them, about no one else than a_c having or wearing the hat and about a_c's incapability to unblock e_{BC_1}.

$$K_{a_b} = W \setminus \Big\{\Big\langle\langle r_{ag},\langle a_a\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{has},\langle a_a,e_{sw}\rangle\rangle,\rho_1\Big\rangle, \tag{52}$$
$$\Big\langle\langle r_{pos},\langle a_a,e_A\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{th},\langle e_{sw}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{wa},\langle e_{sw}\rangle\rangle,\rho_1\Big\rangle,$$
$$\Big\langle\langle r_{ws},\langle a_a,e_{sw}\rangle\rangle,\rho_1\Big\rangle\Big\}\cup\Big\{\Big\langle\langle r_{blk},\langle e_{AB}\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{blk},\langle e_{BC_1}\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_a,e_B\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_b,e_A\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_b,e_C\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_c,e_A\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_c,e_C\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{has},\langle a_a,e_{hat}\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{has},\langle a_b,e_{hat}\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{cub},\langle a_c,e_{BC_1}\rangle\rangle,\rho_0\Big\rangle\Big\}$$

$$K_{a_c} = W \setminus \Big\{\Big\langle\langle r_{ag},\langle a_a\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{has},\langle a_a,e_{sw}\rangle\rangle,\rho_1\Big\rangle, \tag{53}$$
$$\Big\langle\langle r_{pos},\langle a_a,e_A\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{th},\langle e_{sw}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{wa},\langle e_{sw}\rangle\rangle,\rho_1\Big\rangle,$$
$$\Big\langle\langle r_{ws},\langle a_a,e_{sw}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{has},\langle a_b,e_{ring}\rangle\rangle,\rho_1\Big\rangle,\Big\langle\langle r_{th},\langle e_{ring}\rangle\rangle,\rho_1\Big\rangle,$$
$$\Big\langle\langle r_{wa},\langle e_{ring}\rangle\rangle,\rho_1\Big\rangle\Big\}\cup\Big\{\Big\langle\langle r_{blk},\langle e_{AB}\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{blk},\langle e_{BC_1}\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_a,e_B\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_b,e_A\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_b,e_C\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{pos},\langle a_c,e_A\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{pos},\langle a_c,e_C\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{has},\langle a_a,e_{hat}\rangle\rangle,\rho_0\Big\rangle,$$
$$\Big\langle\langle r_{has},\langle a_b,e_{hat}\rangle\rangle,\rho_0\Big\rangle,\Big\langle\langle r_{cub},\langle a_c,e_{BC_1}\rangle\rangle,\rho_0\Big\rangle\Big\}$$

The definition of the agents A consists, apart from the knowledge bases, only of empty sets, as I will not make use of the base of beliefs and obligations or questions are lacking at the initial state of this game.

$$A = \left\{\langle a_a, K_{a_a}, \emptyset, \emptyset, \emptyset\rangle, \langle a_b, K_{a_b}, \emptyset, \emptyset, \emptyset\rangle, \langle a_c, K_{a_c}, \emptyset, \emptyset, \emptyset\rangle\right\} \tag{54}$$

4.5 Operations

There are seven operations of which the following four represents speech acts or combinations of speech acts (see (Searle 1969, chapter 2)):

- o_{comm} stands for the act of asking someone to do something. This can be interpreted in the range from begging to ordering or commanding and corresponds to Searle's class of *directives* (see (Searle 1976, p. 12)). For my setup, politeness or command hierarchy are no objects, yet they could easily be modelled with the available means.

- o_{prom} is the act of asking someone to do something united with giving the promise to do something else in return[26], i.e. a conditioned commitment to the execution of an operation. The two parts of this operation correspond to Searle's *directives* and *commissives* (see (ibid., p. 12)), respectively.

- o_{quest} means the act of posing a question to some agent. Content of the question is a predicate whose truth value is unknown to the questioner[27], hence the operation is restricted to polar questions[28]. Given that questions are nothing else than requests to the addressee of the question to give an answer[29], the question operation belongs as well to the *directives*. The reason for treating questions separately is that, when requiring someone to do something, the operation and parameter to be performed are completely specified[30], while questions intrinsicly

[26] As no actor in my setup will renege on a promise, the temporal order is not an issue here.

[27] According to the *cooperative principle* postulated by Grice (1975), asking already know facts is not a valid behaviour.

[28] Since in my model the answer will originate from the addresse's knowledge base, 'unknown' is a possible truth value to be returned, too. Extending the setup by incorporating the base of beliefs would broaden the set of possible return values even more, so that I should rather refer to this operation as 'truthe value question' than 'polar question'. Cf. (Searle 1969, pp. 31–33).

[29] Complying with the *maxim of qualitiy* by Grice (1975), the addressee is forced to say the truth or admit not to know it, when he decides to answer the question.

[30] This applies only to my setup. Real world examples with multiple to infinitely possible examples of operations and parameters to fulfill the request are easy to cite.

cannot specify the exact resulting parameter tuple, but a set of possible ones.[31]

- o_{assert} is to state a proposition, i.e. a predicate together with a truth value. This operation represents Searle's *assertives*[32].

The operation o_{give} – although not being a speech act – incorporates two agents and composes like this the group of interaction acts together with the speech acts listed above. The remaining operations o_{use} and o_{unbl}, referring to taking a pathway to get to another location and unblocking a blocked pathway, respectively, can be performed by one sole agent and thus are non–interactive acts.

Since the pathways in the setup that I present here are always connections between two locations, the expected result of executing the operation o_{use} is the operator being at the other end's location of the pathway afterwards. The definition in (55) states that the operation needs two parameters apart from the operator. Their assigned meaning is the pathway to be taken and the location of departure.

The preconditions required for the operation being allowed to be executed are that the parameter types match (pathway as well as location), that the pathway is not blocked, that the agent finds itself at the given location and that the location forms part of the pathway. As a result, the agent's being at his initial location becomes false while being at the other location forming part of the pathway becomes true.

$$
\begin{aligned}
o_{use} = \Big\langle & l_{use}, \hat{m}_{\Delta}(l_{use}, \alpha, \langle\beta,\gamma\rangle), \dot{\rho}_W(\langle r_{pw}, \langle\beta\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{loc}, \langle\gamma\rangle\rangle) \\
& \wedge \neg\dot{\rho}_W(\langle r_{blk}, \langle\beta\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{pos}, \langle\alpha,\gamma\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{po}, \langle\beta,\gamma\rangle\rangle), \\
& \{\langle\emptyset, \top, \{W \leftarrow\!\Diamond^{\rho_0}_{(r_{pos},(\alpha,\gamma))}\}\rangle, \\
& \langle\{\delta\}, \dot{\rho}_W(\langle r_{loc}, \langle\delta\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{po}, \langle\beta,\delta\rangle\rangle) \wedge \alpha \neq \delta, \{W \leftarrow\!\Diamond^{\rho_1}_{(r_{pos},(\alpha,\delta))}\}\rangle\}\Big\rangle
\end{aligned}
\tag{55}
$$

Unblocking a blocked pathway can be done by executing the operation o_{unbl} as defined in (56). The only parameter needed is the pathway to be treated. Besides the parameter being a pathway which is blocked, the agent must have the ability to unblock it and be at one location that forms part of

[31] Furthemore, the kind of the expected reaction differs. In case of questions, we expect a statement picking up the question's predicate, while for commands, we want the addressee to act out what the content of the request specifies.

[32] See the synonymical 'representatives' in (Searle 1976, p. 11).

the pathway. The result of a successful execution is the pathway not being blocked anymore.

$$
\begin{aligned}
o_{unbl} = \Big\langle & l_{unbl}, \hat{m}_{\Delta}(l_{unbl}, \alpha, \langle\beta\rangle), \dot{\rho}_W(\langle r_{pw}, \langle\beta\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{blk}, \langle\beta\rangle\rangle) \\
& \wedge \rho_W(\langle r_{cub}, \langle\alpha,\beta\rangle\rangle) \wedge \exists\gamma\colon (\dot{\rho}_W(\langle r_{loc}, \langle\gamma\rangle\rangle) \wedge \dot{\rho}_W(\langle r_{po}, \langle\beta,\gamma\rangle\rangle) \\
& \wedge \dot{\rho}_W(\langle r_{pos}, \langle\alpha,\gamma\rangle\rangle)), \{\langle\emptyset, \top, \{W \leftarrow\!\Diamond^{\rho_0}_{\langle r_{blk},\langle\beta\rangle\rangle}\}\rangle\}\Big\rangle
\end{aligned}
\tag{56}
$$

The operation of giving something to an other agent is realized by o_{give} defined in (57) which takes the receiving agent and the thing to be transferred to him as parameter. Apart from the second parameter's property to be a thing, preconditions are that giver and receiver are distinct, that the giver owns the thing to be given and that both agents are at the same location.[33] Provided that preconditions match, its consequences are that the giver's possession of the given thing becomes false while the receiver's becomes true and that both agents come to know that. Furthermore, the giver wearing the thing is set to false (in terms of the game: in case he was wearing the thing, he doffs it before handing it over).[34]

$$
\begin{aligned}
o_{give} = \Big(& l_{give}, \hat{m}_{\Delta}(l_{give}, \alpha, (\beta,\gamma)), \dot{\rho}_W((r_{th}, (\gamma))) \wedge \alpha \neq \beta \\
& \wedge \dot{\rho}_W((r_{has}, (\alpha,\gamma))) \wedge \exists\delta\colon (\dot{\rho}_W((r_{loc}, (\delta))) \\
& \wedge \dot{\rho}_W((r_{pos}, (\alpha,\delta))) \wedge \dot{\rho}_W((r_{pos}, (\beta,\delta)))), \\
& \{(\emptyset, \top, \{W \leftarrow\!\Diamond^{\rho_0}_{(r_{has},(\alpha,\gamma))}, W \leftarrow\!\Diamond^{\rho_1}_{(r_{has},(\beta,\gamma))}, W \leftarrow\!\Diamond^{\rho_0}_{(r_{ws},(\alpha,\gamma))}, \\
& K_\alpha \leftarrow\!\Diamond^{\rho_0}_{(r_{has},(\alpha,\gamma))}, K_\alpha \leftarrow\!\Diamond^{\rho_1}_{(r_{has},(\beta,\gamma))}, \\
& K_\beta \leftarrow\!\Diamond^{\rho_0}_{(r_{has},(\alpha,\gamma))}, K_\beta \leftarrow\!\Diamond^{\rho_1}_{(r_{has},(\beta,\gamma))}\})\}\Big)
\end{aligned}
\tag{57}
$$

The *directive* class' speech act o_{comm} as in (58) needs as second parameter – besides the addressee – an action that the speaker wants the addressee to do. In this respect, it is necessary that the addressee is the same as the actions's performer, since a directive needs the addressee to be able to carry out the request in order to be valid ('happy' in accordance with Austin (1962, 116 ff.)). Other preconditions are that the same action has not been required to the addressee by the speaker before and that the addressee did also not

[33]The basic conditions for any situation in which an interaction can take place are that both involved agents share the same location and that they are not identical. I will assume that as known and omit it for further interactive operations.

[34]No other agent at the same location will observe the change of ownership. However, doffing something will trigger an observation of the other agents being there.

commit himself to do it toward the speaker yet. If all conditions match, the action will be added to both agents' obligations set, without declaring no action to be performed by the speaker in return.

$$
\begin{aligned}
o_{comm} = \Big(&l_{comm}, \hat{m}_{\Delta}(l_{comm}, \alpha, (\beta, \gamma)), \exists l_o \exists \psi \colon (\gamma = (l_o, \beta, \psi)) \\
&\wedge \alpha \neq \beta \wedge \exists \delta \colon (\dot{\rho}_W((r_{loc}, (\delta))) \wedge \dot{\rho}_W((r_{pos}, (\alpha, \delta))) \\
&\wedge \dot{\rho}_W((r_{pos}, (\beta, \delta)))) \wedge \neg(\exists \varsigma \colon ((\alpha, \beta, \varsigma, \gamma) \in X_\alpha) \\
&\vee \exists \varsigma \colon ((\beta, \alpha, \gamma, \varsigma) \in X_\alpha)), \{(\emptyset, \top, \{X_\alpha \leftarrow\bigcirc^{\beta\leftarrow\alpha}_{\gamma\leftarrow ?}, X_\beta \leftarrow\bigcirc^{\beta\leftarrow\alpha}_{\gamma\leftarrow ?}\})\}\Big)
\end{aligned}
\tag{58}
$$

The complement to o_{comm} is o_{prom} which is specified in (59). Consisting besides a command as in (58) of commissive part too, an additional parameter is needed to contain that action. Both actions' parameters need to match the relative agents. In contrast to the addressee being the operator of the directive part, the speaker needs to match the commissive part's one (again following Searle's requirements to speech acts).

In my definition of making a conditioned commitment, the speaker must have been asked (or commanded, ordered and so forth) to do something before he becomes able to promise to do it if the solicitant is willing to perform another action. Therefore, a matching tuple must exist in the speaker's obligations set. On the other hand, the same promise must not have been given in combination with the action required for him to perform, nor may an obligation exist for the same case with exchanged roles.

As consequence to this operation, the required action will be added to both agents' obligations set using the second action parameter as return.

$$
\begin{aligned}
o_{prom} = \Big(&l_{prom}, \hat{m}_{\Delta}\Big(l_{prom}, \alpha, (\beta, \gamma, \delta)), \exists l_o \exists \psi \colon (\alpha = (l_o, \gamma, \psi)) \\
&\wedge \exists l_o \exists \psi \colon (\beta = (l_o, \delta, \psi)) \wedge \alpha \neq \beta \wedge \exists \epsilon \colon (\dot{\rho}_W((r_{loc}, (\epsilon))) \\
&\wedge \dot{\rho}_W((r_{pos}, (\alpha, \epsilon))) \wedge \dot{\rho}_W((r_{pos}, (\beta, \epsilon)))) \wedge \exists \varsigma \colon ((\beta, \alpha, \varsigma, \gamma) \in X_\alpha) \\
&\wedge \neg(\exists \varsigma : ((\beta, \alpha, \delta, \gamma) \in X_\alpha) \vee \exists \varsigma \colon ((\alpha, \beta, \gamma, \delta) \in X_\alpha)), \\
&\{(\emptyset, \top, \{X_\alpha \leftarrow\bigcirc^{\beta\leftarrow\alpha}_{\delta\leftarrow\gamma}, X_\beta \leftarrow\bigcirc^{\beta\leftarrow\alpha}_{\delta\leftarrow\gamma}\})\}\Big)
\end{aligned}
\tag{59}
$$

The operation to pose a question to an agent, o_{quest} defined in (60), consist besides the addressed agent of a predicate whose assigned truth value is object of the inquirer's demand. The only condition, in addition to the standard interaction ones, is that the same question is not 'open', i.e. the question does not exist in the agent's question set without containing a result truth value. As consequence to matching preconditions, the question will be added to both agent's question sets.

$$o_{quest} = \Big(l_{quest}, \hat{m}_{\Delta}(l_{quest}, \alpha, (\beta, \gamma)), \alpha \neq \beta \tag{60}$$
$$\wedge\ \exists\delta\colon (\dot{\rho}_W((r_{loc}, (\delta))) \wedge \dot{\rho}_W((r_{pos}, (\alpha, \delta))) \wedge \dot{\rho}_W((r_{pos}, (\beta, \delta))))$$
$$\wedge\ ((\alpha, \beta, \gamma, \varnothing) \notin Q_\alpha), \{(\emptyset, \top, \{Q_\alpha \leftarrow\Box^{\beta\leftarrow\alpha}_{\gamma=?}, Q_\beta \leftarrow\Box^{\beta\leftarrow\alpha}_{\gamma=?}\})\}\Big)$$

Questions can be responded by asserting using the operation o_{assert} as defined in (61). In addition to o_{quest}, a truth value is needed to turn the predicate into a proposition. In order to answer a question, it has to be asked before[35] (that is given by a corresponding tuple in the questions set) and no answer has to be given yet (indicated by no matching tuple in the set containing a truth value). The consequence of this operation is that the predicate as well as the stated truth value will be added to both agents' question set as well as the proposition to the addressee's knowledge base[36].

$$o_{assert} = \Big(l_{assert}, \hat{m}_{\Delta}(l_{assert}, \alpha, (\beta, \gamma, \tau)), \alpha \neq \beta \tag{61}$$
$$\wedge\ \exists\delta\colon (\dot{\rho}_W((r_{loc}, (\delta))) \wedge \dot{\rho}_W((r_{pos}, (\alpha, \delta))) \wedge \dot{\rho}_W((r_{pos}, (\beta, \delta))))$$
$$\wedge\ ((\alpha, \beta, \gamma, \varnothing) \in Q_\alpha) \wedge \nexists\rho \in L_\rho\colon ((\beta, \alpha, \gamma, \rho) \in Q_\alpha),$$
$$\{(\emptyset, \top, \{Q_\alpha \leftarrow\Box^{\beta\leftarrow\alpha}_{\gamma=\tau}, Q_\beta \leftarrow\Box^{\beta\leftarrow\alpha}_{\gamma=\tau}, K_\beta \leftarrow\Diamond^{\tau}_{\gamma}\})\}\Big)$$

4.6 Triggers

Seven triggers ensure that the agents observe their environment (t_1 to t_4) and provide them with simple external deductive reasoning mechanisms (t_5 to t_7 and partly also t_2). The triggers as a whole can be seen as the non player–controlled part of the game mechanics.

The trigger t_1 defined in (62) embodies the observation of any agent being at a location for which the truth value of a particular agent's being there changes (i.e. either the start or the end position of a change of location). This does also affect the acting agent itself.

$$t_1 = \Big(l_1, W\widehat{\leftarrow\Diamond}^{\tau}_{(r_{pos},(\alpha,\beta))}, \top, \tag{62}$$
$$\{(\{\gamma\}, \dot{\rho}_W((r_{ag}, (\gamma))) \wedge (\dot{\rho}_W((r_{pos}, (\gamma, \beta))) \vee \gamma = \alpha), \{K_\gamma \leftarrow\Diamond^{\tau}_{(r_{pos},(\alpha,\beta))}\})\}\Big)$$

[35] Agents have no means to just state something they find important for an other agent to know, since without beliefs, they cannot judge about the other ones' knowledge which is needed for not to violate the Gricean Maxim of Quantity (see (Grice 1975)).

[36] According with the Gricean Maxim of Quality (see (ibid.)), the speaker is bound to state merely propositions he knows for sure (i.e. propositions contained by his knowledge base). Hence, the addressee can take them for granted to be true.

The becoming true of an agent's being at one location is matched by t_2 defined in (63). The agent (= newcomer) included in the corresponding predicate comes to know the current location (he might not have known where the pathway he used connects to) and the being there of any agent for which this applies. Every agent at that location finds out about the existence of the newcomer and vice versa. Likewise, they observe what they other agents are wearing and, if they do wear something, they conclude that it must be a thing that is wearable. For every pathway connected to the location matched by the trigger, the also matched agent finds out which pathways connect there and which of them are blocked.

The last three consequences concern the reasoning of the agents: The newcomer deduces that while being at the matched location, his being at any other location must be false. Comparably, any agent being there deduces that said matched agent is at no other location and vice versa, while other agents not being there cannot know the truth value of the matched agent being at the matched location and vice versa.

$$
\begin{aligned}
t_2 = \big(& l_2, W\widehat{\leftarrow\Diamond}^{\rho_1}_{(r_{pos},(\alpha,\beta))}, \top, \\
& \{(\emptyset, \top, \{K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{loc},(\beta))}\}), \\
& (\{\gamma\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))), \{K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{pos},(\gamma,\beta))}\}), \\
& (\{\gamma\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \gamma \neq \alpha, \\
& \{K_\gamma \leftarrow\Diamond^{\rho_1}_{(r_{ag},(\alpha))}, K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{ag},(\gamma))}\}), \\
& (\{\gamma,\delta\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \dot\rho_W((r_{wa},(\delta))) \wedge \gamma \neq \alpha, \\
& \{K_\gamma \leftarrow\Diamond^{W}_{(r_{ws},(\alpha,\delta))}, K_\alpha \leftarrow\Diamond^{W}_{(r_{ws},(\gamma,\delta))}\}), \\
& (\{\gamma,\delta\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \dot\rho_W((r_{wa},(\delta))) \\
& \wedge \dot\rho_W((r_{ws},(\alpha,\delta))) \wedge \gamma \neq \alpha, \{K_\gamma \leftarrow\Diamond^{\rho_1}_{(r_{th},(\delta))}, K_\gamma \leftarrow\Diamond^{\rho_1}_{(r_{wa},(\delta))}\}), \\
& (\{\gamma,\delta\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \dot\rho_W((r_{wa},(\delta))) \\
& \wedge \dot\rho_W((r_{ws},(\gamma,\delta))) \wedge \gamma \neq \alpha, \{K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{th},(\delta))}, K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{wa},(\delta))}\}), \\
& (\{\gamma\}, \dot\rho_W((r_{pw},(\gamma))) \wedge \dot\rho_W((r_{po},(\gamma,\beta))) \\
& \{K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{pw},(\gamma))}, K_\alpha \leftarrow\Diamond^{\rho_1}_{(r_{po},(\gamma,\beta))}, K_\alpha \leftarrow\Diamond^{W}_{(r_{blk},(\gamma))}\}), \\
& (\{\gamma\}, \dot\rho_W((r_{loc},(\gamma))) \wedge \gamma \neq \beta, \{K_\alpha \leftarrow\Diamond^{\rho_0}_{(r_{pos},(\alpha,\gamma))}\}), \\
& (\{\gamma,\delta\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \dot\rho_W((r_{loc},(\delta))) \\
& \wedge \gamma \neq \alpha \wedge \delta \neq \beta, \{K_\alpha \leftarrow\Diamond^{\rho_0}_{(r_{pos},(\gamma,\delta))}, K_\gamma \leftarrow\Diamond^{\rho_0}_{(r_{pos},(\alpha,\delta))}\}), \\
& (\{\gamma,\delta\}, \dot\rho_W((r_{ag},(\gamma))) \wedge \neg\dot\rho_W((r_{pos},(\gamma,\beta))) \wedge \dot\rho_W((r_{loc},(\delta)))
\end{aligned}
\tag{63}
$$

$$\wedge\, \gamma \neq \alpha \wedge \delta \neq \beta, \{K_\alpha \leftarrow\!\Diamond^{\rho_u}_{(r_{pos},(\gamma,\delta))}, K_\gamma \leftarrow\!\Diamond^{\rho_u}_{(r_{pos},(\alpha,\delta))}\})\}\Big)$$

When an agent dons or doffs a wearable thing, the truth value for him wearing that thing changes and thus triggers t_3 as described in (64). The knowledge base of all agents being at the same location will be updated with this proposition.

$$t_3 = \Big(l_3, W\widehat{\leftarrow\!\Diamond}^{\tau}_{(r_{ws},(\alpha,\beta))}, \top, \{(\{\gamma,\delta\}, \dot{\rho}_W((r_{loc},(\gamma))) \wedge \dot{\rho}_W((r_{pos},(\alpha,\gamma))) \\ \wedge\, \dot{\rho}_W((r_{ag},(\delta))) \wedge \dot{\rho}_W((r_{pos},(\delta,\gamma))), \{K_\delta \leftarrow\!\Diamond^{\tau}_{(r_{ws},(\alpha,\beta))}\})\}\Big) \tag{64}$$

In the same way, the trigger t_4 as defined in (65) updates the agents' knowledge base when a pathway becomes unblocked.

$$t_4 = \Big(l_4, W\widehat{\leftarrow\!\Diamond}^{\rho_0}_{(r_{blk},(\alpha))}, \top, \{(\{\beta,\gamma\}, \dot{\rho}_W((r_{loc},(\beta))) \wedge \dot{\rho}_W((r_{po},(\alpha,\beta))) \\ \wedge\, \dot{\rho}_W((r_{ag},(\gamma))) \wedge \dot{\rho}_W((r_{pos},(\gamma,\beta))), \{K_\gamma \leftarrow\!\Diamond^{\rho_0}_{(r_{blk},(\alpha))}\})\}\Big) \tag{65}$$

The trigger t_5 defined in (66) stands for the agent reasoning that someone who wears something must own it too and that no other agent can wear or own it at the same time.

$$t_5 = \Big(l_5, K_\vartheta\widehat{\leftarrow\!\Diamond}^{\tau}_{(r_{ws},(\alpha,\beta))}, \top, \{(\emptyset, \top, \{K_\vartheta \leftarrow\!\Diamond^{\rho_1}_{(r_{has},(\alpha,\beta))}\}), \\ (\{\gamma\}, \dot{\rho}_W((r_{ag},(\gamma))) \wedge \gamma \neq \alpha\{K_\gamma \leftarrow\!\Diamond^{\rho_0}_{(r_{ws},(\gamma,\beta))}, K_\gamma \leftarrow\!\Diamond^{\rho_0}_{(r_{has},(\gamma,\beta))}\})\}\Big) \tag{66}$$

When an agent comes to know that a location is part of a pathway and he knows another location for which the same applies, he deduces in t_6 defined in (67) that any other location apart from these two does not form part of the pathway.

$$t_6 = \Big(l_6, K_\vartheta\widehat{\leftarrow\!\Diamond}^{\rho_1}_{(r_{po},(\alpha,\beta))}, \top, \{(\{\gamma,\delta\}, \dot{\rho}_W((r_{loc},(\gamma))) \wedge \dot{\rho}_W((r_{loc},(\delta))) \\ \wedge\, \dot{\rho}_{K_\vartheta}((r_{po},(\alpha,\gamma))) \wedge \gamma \neq \beta \wedge \delta \neq \gamma \wedge \delta \neq \beta, \{K_\vartheta \leftarrow\!\Diamond^{\rho_0}_{(r_{po},(\alpha,\delta))}\})\}\Big) \tag{67}$$

Finally, if an agent comes to know in t_7 defined in (68) that a thing is owned by one agent, he deduces that no other agent can be the owner.

$$t_7 = \Big(l_7, K_\vartheta\widehat{\leftarrow\!\Diamond}^{\rho_1}_{(r_{has},(\alpha,\beta))}, \top, \{(\{\gamma\}, \dot{\rho}_W((r_{ag},(\gamma))) \wedge \gamma \neq \alpha, \\ \{K_\vartheta \leftarrow\!\Diamond^{\rho_0}_{(r_{has},(\gamma,\beta))}\})\}\Big) \tag{68}$$

4.7 Actors

In the stage concerning setup as described in the preceding sections, the 'physical' situation as well as the 'mental' one in terms of knowledge bases have already been defined. In the following, the other part of the mental configuration regarding personal attitudes will be defined. Since in this setup all agents are artificial, any actor $\vartheta \in \Theta$ comprises a set $\hat{A}_a$ of agent related values consisting of pairs of agents a_i and their mental representations $\hat{a}_j$. Every mental representation is a 4–tuple:

$$\hat{a} = \langle \vec{p}, \vec{s}, \Xi, \omega \rangle \tag{69}$$

where $\vec{p}$ and $\vec{s}$ are vectors standing for the agent's personality, Ξ is a set of triples of an event pattern $\hat{e}$, conditions $\breve{c}$ and a shift vector $\vec{s}_\Delta$ that describes which influence the particular event will have on the particular agent and ω is a function to calculate the well–being of an agent given $\vec{p}$ and $\vec{s}$.

The actors are stuffed with individual and shared preferences for propositions, or rather truth value changes of propositions, as well as for the performance of actions. For modeling preferences, a *personality vector* is assigned to every actor, i.e. a unit vector in the vector space spanned by the personality dimensions included. In the case of using the traits from the 'Big Five' theory (as defined by Costa and McCrae (1992)), the vector space would be five–dimensional. For my setup, I only use two factors – more or less correlated to materialistic (first dimension) or idealistic (second dimension) attitudes. Things that can be owned carry a materialistic value in most cases (although an idealistic value may exists independently for a particular actor), while the act of answering a question, for instance, comes solely with an idealistic value.

Every actor possesses, apart from the personality vector $\vec{p}$, another vector $\vec{s}$ in the same vector spane representing his current *mental state*. His well–being ω is then defined as the inverse sine of the angle between these two vectors and the length of the latter one:

$$\omega = |\vec{s}| \cdot \arcsin\left(\frac{\vec{b} \cdot \vec{s}}{|\vec{s}|}\right) \tag{70}$$

Thus, increasing the angle between $\vec{p}$ and $\vec{s}$ lowers the well–being ω as well as decreasing the length of $\vec{s}$. The well–being becomes zero, when either $\vec{s}$ equals to $\vec{0}$ or the angle between the vectors becomes 90°.

In order to evaluate the outcome of performing a particular action, the current well–being value has to be compared to the updated one after the

action has been performed. In my definition, actors will not do anything that impairs their state, i.e. lowers their well–being value.[37] In this way, decisions in order to perform a particular action or not depend on the stage, on the considered actor's personality vector as well as on his current 'mental' state. The same actor in the same situation might come to another decision if his personality vector is inclined more towards the materialistic or idealistic basis vector. The same applies to his mental state.

For reasons of convenience, each of the three agents defined in section 4.4 gets assigned one actor turning the pairs agent–actor into complete artificial characters.[38] The actor playing an agent can be inquired via the function f_ϑ as defined in (27):

$$f_\vartheta(a_a) = \vartheta_1 \qquad f_\vartheta(a_b) = \vartheta_2 \qquad f_\vartheta(a_c) = \vartheta_3 \tag{71}$$

In this setup, the set $\hat{A}_a$ of each ϑ_1, ϑ_1 and ϑ_1 contains exactly one agent related tuple:

$$\begin{aligned} \hat{A}_{\vartheta_1} &= \{\langle a_a, \langle \vec{p}_{a_a}, \vec{s}_{a_a}, \Xi_{a_a}, \omega \rangle\rangle\} \\ \hat{A}_{\vartheta_2} &= \{\langle a_b, \langle \vec{p}_{a_b}, \vec{s}_{a_b}, \Xi_{a_b}, \omega \rangle\rangle\} \\ \hat{A}_{\vartheta_2} &= \{\langle a_c, \langle \vec{p}_{a_c}, \vec{s}_{a_c}, \Xi_{a_c}, \omega \rangle\rangle\} \end{aligned} \tag{72}$$

The initial personality vectors are set with the one belonging to a_b being the most materialistic, a_c the most idealistic and a_a in an indeclined position:

$$\vec{p}_{a_a} = R(45°)\begin{pmatrix}1\\0\end{pmatrix} \qquad \vec{p}_{a_b} = R(30°)\begin{pmatrix}1\\0\end{pmatrix} \qquad \vec{p}_{a_c} = R(60°)\begin{pmatrix}1\\0\end{pmatrix} \tag{73}$$

The initial mental states are set in such way that a_a starts with a convenient state, while the other two have a smaller scope and, hence, theoretically fewer options for decisions:

$$\vec{s}_{a_a} = \begin{pmatrix}15\\15\end{pmatrix} \qquad \vec{s}_{a_b} = \begin{pmatrix}3\\3\end{pmatrix} \qquad \vec{s}_{a_c} = \begin{pmatrix}3\\3\end{pmatrix} \tag{74}$$

In order to be able to make use of variables bound by the event pattern, the labels must be converted into numerics. This is done for truth values by the function ι:

$$\iota(\tau) = \begin{cases} 1 & \text{if } \tau = \rho_1 \\ -1 & \text{if } \tau = \rho_0 \\ 0 & \text{otherwise} \end{cases} \tag{75}$$

[37] Aside from an actor's ignorance of a world state fact accounting for an impairment to his well–being.

[38] The option of replacing one or more of the coded actors by human players remains, though.

If not used for the vector, any variable disposed by the event pattern and thus universally quantified is just seen as place holder.

Ξ_{a_a} contains the shift vectors for the agent a_a that will apply if the respective pattern matches and the respective condition holds is set as follows:

$$\begin{aligned}\Xi_{a_a} = \Big\{ & (K_{a_a} \widehat{\leftarrow\diamond}^{\tau}_{(r_{pos},(a_a,e_A))}, \top, \iota(\tau)\begin{pmatrix}4 & 0\end{pmatrix}^T), \\ & (K_{a_a} \widehat{\leftarrow\diamond}^{\tau}_{(r_{pos},(a_a,e_B))}, \top, \iota(\tau)\begin{pmatrix}6 & 0\end{pmatrix}^T), \\ & (K_{a_a} \widehat{\leftarrow\diamond}^{\tau}_{(r_{pos},(a_a,e_C))}, \top, \iota(\tau)\begin{pmatrix}\infty & 0\end{pmatrix}^T), \\ & (K_{a_a} \widehat{\leftarrow\diamond}^{\rho_0}_{(r_{pos},(a_b,e_B))}, \top, \begin{pmatrix}0 & 1\end{pmatrix}^T), \\ & (K_{a_a} \widehat{\leftarrow\diamond}^{\rho_0}_{(r_{blk},(e_{BC_1}))}, \top, \begin{pmatrix}3 & 7\end{pmatrix}^T), \\ & (K_{a_a} \widehat{\leftarrow\diamond}^{\tau}_{(r_{has},(a_a,e_{sw}))}, \top, \iota(\tau)\begin{pmatrix}7 & 0\end{pmatrix}^T), \\ & (\hat{e}_{\Delta}(o_{quest}, a_a, \psi), \top, \begin{pmatrix}0 & -1\end{pmatrix}^T), \\ & (\hat{e}_{\Delta}(o_{asst}, a_a, \psi), \top, \begin{pmatrix}0 & 2\end{pmatrix}^T), \\ & (\hat{e}_{\Delta}(\mu, a_a, \psi), (a_a, \eta, \varnothing, \mu) \in X_{a_a}, \begin{pmatrix}0 & 4\end{pmatrix}^T), \\ & (\hat{e}_{\Delta}(\mu, a_a, \psi), (a_a, \eta, \nu, \mu) \in X_{a_a} \wedge \nu \neq \varnothing, \begin{pmatrix}0 & 3\end{pmatrix}^T)\Big\}\end{aligned} \tag{76}$$

In my definitions, the actor ϑ_1 (supposed to be female as well) wants the agent a_a to get to e_C and getting to e_B is yet an advantage for her (what is reflected by a better vector (i.e. a longer vector if positive). Once she has reached e_C, her well–being will be that high, that the gameplay component ends the game. For a not further explained reason, she wants a_b to leave e_B once she comes to know him. Getting the pathway e_{BC_1} unblocked will influence her decision making, when she will have found it being blocked. The sword comes along with a materialistic value that will be withdrawn if she gives it away. Giving a response is linked for her to a positive idealistic value and asking something to a negative one. Complying a request is mentally recompensed higher for her, if it was unconditioned, i.e. the requester did a o_{comm} but no o_{prom} before.

Actor ϑ_2 has no other motivation for a_b than to get the sword once he will be aware of its existence, while ϑ_3 is not provided with any other means for a_c than acting in order to modify his mental state. Both actors share the values for asking and responding with ϑ_3. In matters of doing a requested action, both actors evaluate it higher if the the request was unconditioned.

Compared to ϑ_2, the actor ϑ_3 does not set much store by letting a_c complying requests.

$$\begin{aligned}\Xi_{a_b} = \Big\{&(K_{a_b} \widehat{\leftarrow\diamond}^{\tau}_{(r_{has},(a_b,e_{sw}))}, \top, \iota(\tau)\begin{pmatrix}7 & 2\end{pmatrix}^T), \\ &(\hat{e}_\Delta(o_{quest}, a_b, \psi), \top, \begin{pmatrix}0 & -1\end{pmatrix}^T), \\ &(\hat{e}_\Delta(o_{asst}, a_b, \psi), \top, \begin{pmatrix}0 & 2\end{pmatrix}^T), \\ &(\hat{e}_\Delta(\mu, a_a, \psi), (a_b, \eta, \varnothing, \mu) \in X_{a_b}, \begin{pmatrix}0 & 3\end{pmatrix}^T), \\ &(\hat{e}_\Delta(\mu, a_a, \psi), (a_b, \eta, \nu, \mu) \in X_{a_b} \wedge \nu \neq \varnothing, \begin{pmatrix}0 & 2\end{pmatrix}^T)\Big\}\end{aligned} \tag{77}$$

$$\begin{aligned}\Xi_{a_c} = \Big\{&(\hat{e}_\Delta(o_{quest}, a_c, \psi), \top, \begin{pmatrix}0 & -1\end{pmatrix}^T), \\ &(\hat{e}_\Delta(o_{asst}, a_c, \psi), \top, \begin{pmatrix}0 & 2\end{pmatrix}^T), \\ &(\hat{e}_\Delta(\mu, a_a, \psi), (a_c, \eta, \varnothing, \mu) \in X_{a_c}, \begin{pmatrix}0 & 1\end{pmatrix}^T), \\ &(\hat{e}_\Delta(\mu, a_a, \psi), (a_c, \eta, \nu, \mu) \in X_{a_c} \wedge \nu \neq \varnothing, \begin{pmatrix}0 & 0\end{pmatrix}^T)\Big\}\end{aligned} \tag{78}$$

After an actor gets passed the list of all modifications applying to one of his agents by g_i defined in (42), he calculates the sum over all vectors he could derive from said list. The resulting vector added to his current mental state leads to his new mental state. If that one lies out of the first quadrant (i.e. at least one dimension's value is negative), the operation in question is excluded from the possible actions the actor could do. In case the underlying operation defies his control, out–of–range values are accepted, albeit the actor is only allowed to continue with operations that take his mental state back to the first quadrant.

All these settings given in this section are meant to serve as base for experiments. In real game situations, every single value has to be motivated by a fictitious character profile. One could also imagine actors adjusting their values for certain propositions or actions whilst the game is played. The personality vector on the contrary is meant to remain fixed for the whole game.[39]

4.8 Game master and gameplay

The gameplay and the game master are kept minimalist: After having entered the game in its initial state, control is given to the game master who turns

[39] As stated by McCrae and Costa (2003), personality becomes stable between the age of 20 and 30 and my aim is not to model actors in adolescence.

it over to all agents sequentially starting over at the end. Between adjacent actors' turns, when control is given back to the gameplay, end conditions of any actor and the game itself are checked in order to terminate if any of them matches. The gameplay is depicted by figure 4.

This procedure, which does not take into account any of the possibilities presented in section 3.3, assumedly would result in quite unnatural interactions. As my setup merely defines three agents, of which one primarily serves as supporting role, the lack of 'intelligence' of the game master does not stand out.

4.9 Sample game traces

The setup described in the sections 4.1 to 4.8 provides a basis for my implementation. For purposes of readability, the labels in the following are replaced in figures and descriptions by more memorable names, namely *Alice* (for a_a), *Bob* (for a_b), *Carlos* (for a_c), *Amonlond* (for e_A), *Baernsari* (for e_B), *Caradcalen* (for e_C), *forest track* (for e_{AB}), *river bridge way* (for e_{BC_1}), *old dune trail* (for e_{BC_2}), *sword* (for e_{sw}), *hat* (for e_{hat}) and *ring* (for e_{ring}). Other labels will be assigned names when they are used in the context of text generation.

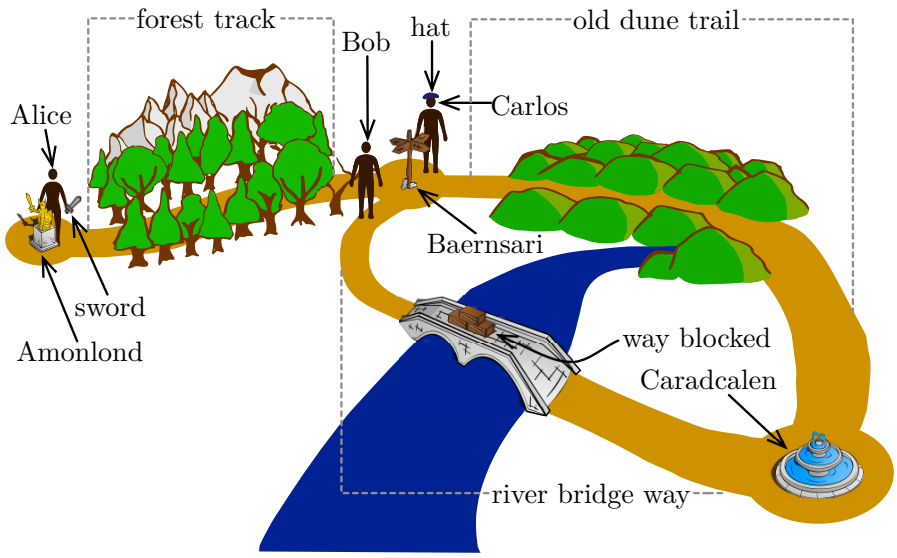

Figure 5: The game world from Figure 3 described by names instead of labels.

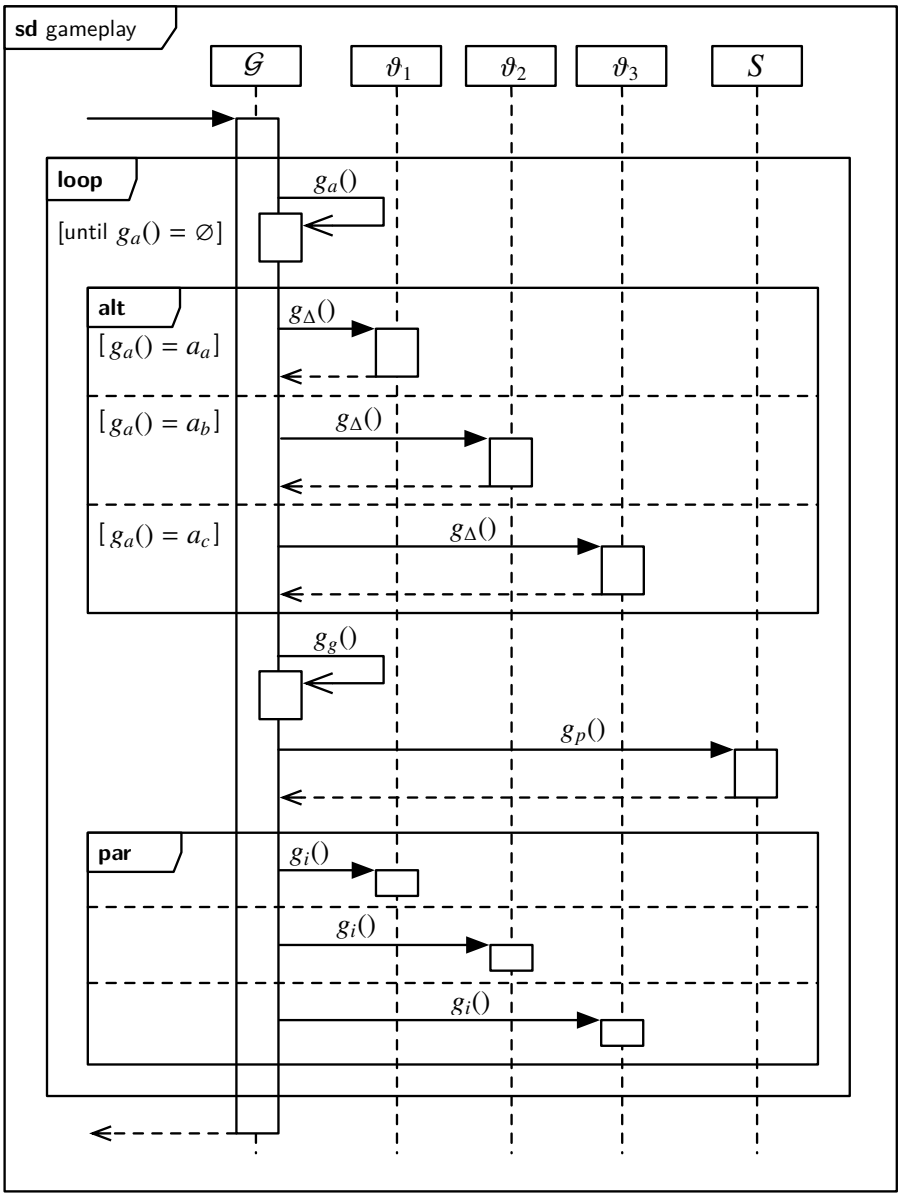

Figure 4: The gameplay for the setup of chapter 4.

Figure 5 shows the initial situation of the particular setup of chapter 4. Alice stands with the sword in her hand in Amonlond, which is connected to Baernsari via the forest track. There, both Bob and Carlos reside with Carlos wearing a hat. Two paths, the river bridge way and the old dune trail, connect Baernsari with Caradcalen. The river bridge way is blocked by an obstacle.

The agents' knowledge about the world is a subset of the world state, i.e. the totality of every fact constituting the world, thus agents base their decisions on the assumption that their knowledge about the world corresponds to real world. Figure 6 depicts the world as the agent Alice knows it.

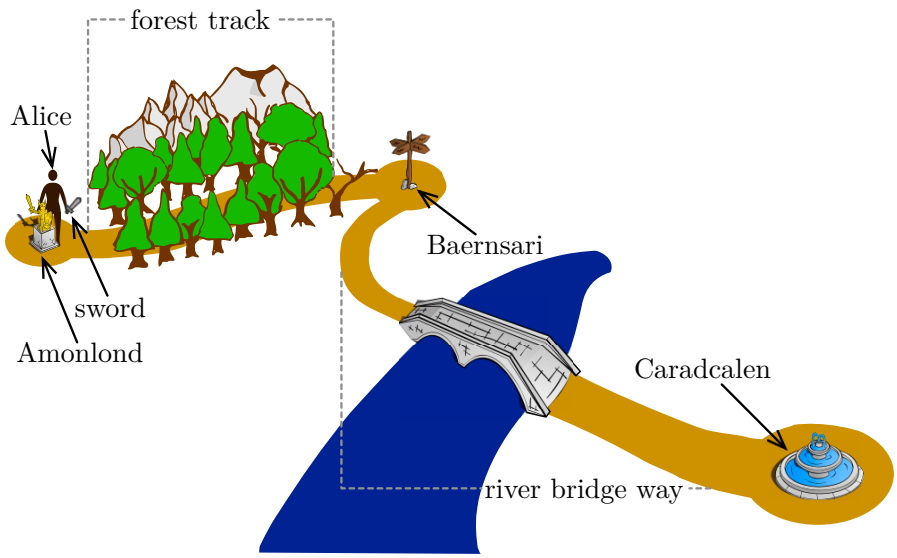

Figure 6: The part of the world known to Alice at the initial state.

As o_{use} and o_{unbl} (*use* and accordingly *unblock* in the following) are the only actions that a lone agent can perform and there is nothing unblockable in Amonlond, the single remaining alternative for Alice is to go to Baernsari, provided that performing this action – to the best of her knowledge – would not affect her well–being in a negative way. Deriving consequences of that action taking her knowledge as a basis leads to the modifications illustrated in Figure 7.

$$(o_{use}, a_a, (a_a, e_{AB}, e_A)) \xrightarrow{o_{use}} \begin{array}{l} W \leftarrow\!\diamond^{\rho_0}_{pos(a_a,e_A)} \\ W \leftarrow\!\diamond^{\rho_1}_{pos(a_a,e_B)} \end{array} \begin{array}{l} \xrightarrow{t_1} K_a \leftarrow\!\diamond^{\rho_0}_{pos(a_a,e_A)} \\ \xrightarrow{t_1} \\ \xrightarrow{t_2} K_a \leftarrow\!\diamond^{\rho_1}_{pos(a_a,e_B)} \end{array}$$

Figure 7: Alice' expectation of what her using the forest track departing from Amonlond would effect.

The two modifications affecting her knowledge base have to be taken into account in order to calculate the effect performing this action would have on her well–being. As defined in (76), the difference between her being in Baernsari compared to being Amonlond amounts to a positive vector, which means that, given her personality vector from (73) not being orthogonal to the resulting vector, the final result will in any case be positive. Therefore, her first action is to take the forest track to Baernsari.

The real consequences of this action can be reconstructed with the help of Figure 8. The predicted modifications from Figure 7 are contained, but, as Alice did not know about Bob and Carlos, plenty of unanticipated ones add to them. Since Alice has not been equipped with the knowledge about the river bridge way being blocked, she discovers this circumstance once she is in Baernsari. The situation at this particular time turns out to be as depicted in Figure 9.

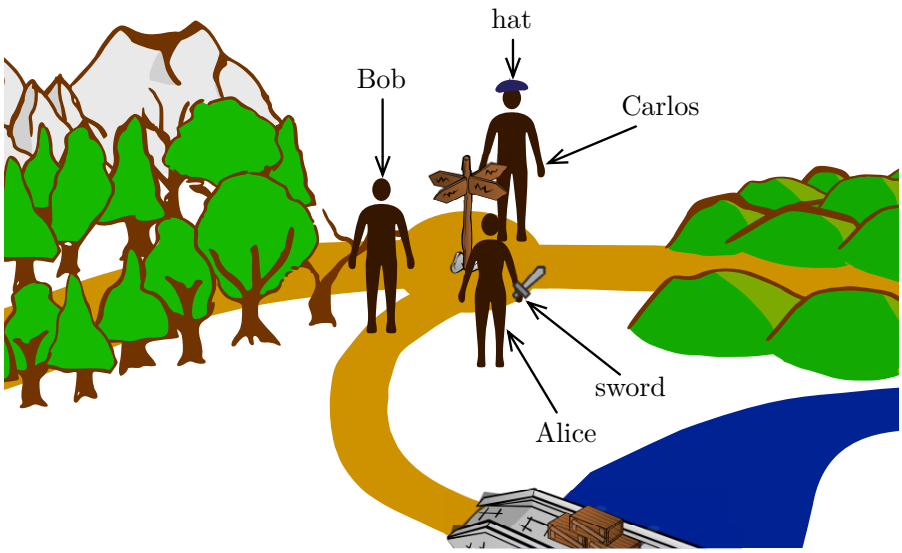

Figure 9: The situation when Alice has arrived at Baernsari.

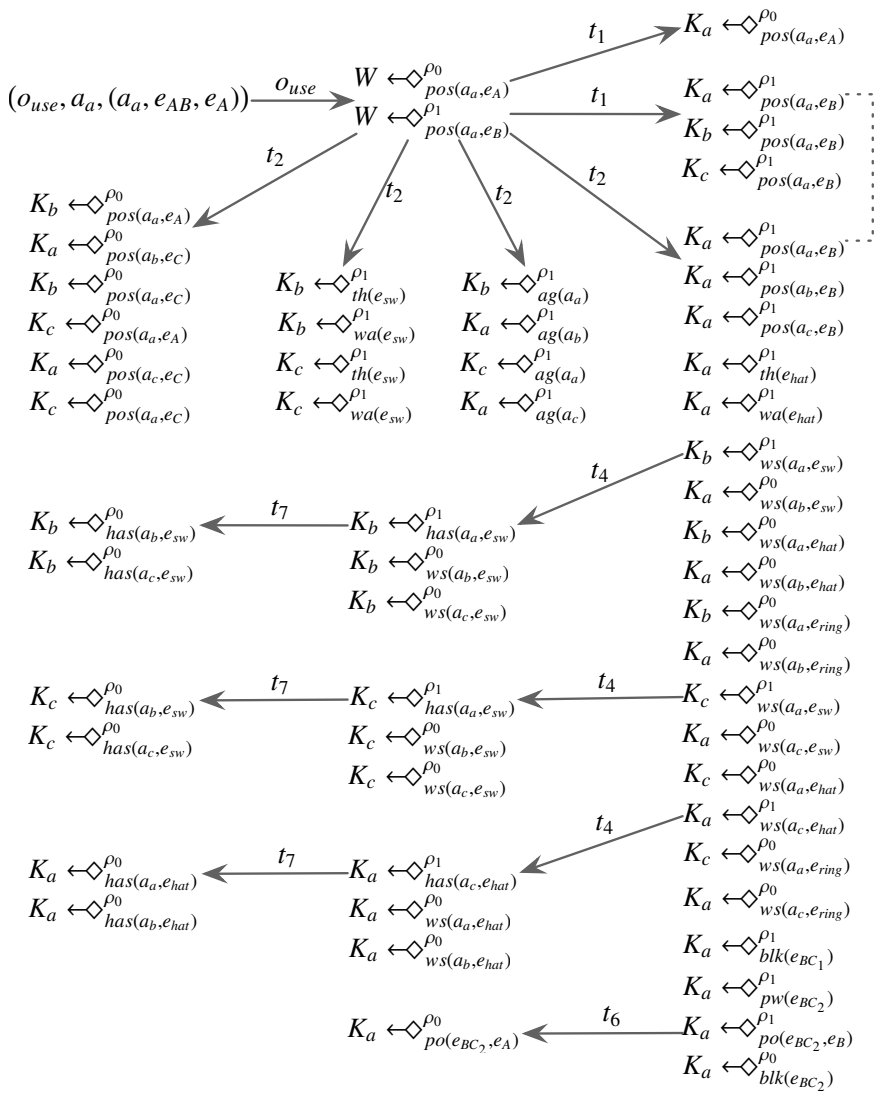

Figure 8: Consequences of the action of Alice using the forest track departing from Amonlond. The arrows are labeled with the respective operation or trigger that causes the modifications toward which the arrow is directed to be applied.

Henceforward, the agents are in the position to perform any kind of interaction among each other. Figure 10 visualizes a script of the course of the game that unfolds when the game is played with the stage set up as specified in section 4.1 to 4.6, the actors as in section 4.7 and the gameplay and game master as in section 4.8. The text is generated directly from the sequence of actions taken by the agents using superficial rules to generate referring expressions and discourse markers.

```
ALICE uses the forest track departing from Amonlond.

                            BOB
                    (to Alice)
             Please give me the sword!

                            ALICE
             Well, can you unblock the
             river bridge way?

                            BOB
             Yes, I can.

                            ALICE
             Please do so!

                            BOB
             Okay, I will do that, if you
             give me the sword.

                            ALICE
             Can Carlos do it?

                            BOB
             No, he can't.

ALICE gives BOB the sword.

As a consequence, BOB unblocks the river bridge way.

ALICE uses the river bridge way departing from Baernsari.
```

Figure 10: Script of what happens when the game is run with the given configuration.

In this scenario, Alices hands over the sword to Bob because getting the river bridge way unblocked is of a greater value to her than loosing the sword. Changing the value Bob ascribes to the sword by redefining Ξ_{a_b} as described in (79) leads to Bob rather being altruistic by unblocking the path for Alice without any reward than expecting her to give him her sword in return for that action. In this vein, Alice arrives at her destination without her well–being being decreased as described by Figure 11.

$$\begin{aligned}\Xi_{a_b} \leftarrow \Xi_{a_b} \setminus (K_{a_b} \widehat{\leftarrow\Diamond}^{\tau}_{(r_{has},(a_b,e_{sw}))}, \top, \iota(\tau)\begin{pmatrix}7 & 2\end{pmatrix}^T) \\ \cup (K_{a_b} \widehat{\leftarrow\Diamond}^{\tau}_{(r_{has},(a_b,e_{sw}))}, \top, \iota(\tau)\begin{pmatrix}1 & 1\end{pmatrix}^T)\end{aligned} \tag{79}$$

```
ALICE uses the forest track departing from Amonlond.

                          BOB
                  (to Alice)
            Please give me the sword!

                          ALICE
            Well, can you unblock the
            river bridge way?

                          BOB
            Yes, I can.

                          ALICE
            Please do so!

BOB unblocks the river bridge way.

ALICE uses the river bridge way departing from Baernsari.
```

Figure 11: Course of the game when (79) replaces Bob's preference entry for him having the sword.

Another variation from the setup is depicted by figure 12, where Carlos comes with the same preferences for the sword as Bob does as specified in (80). Since Alice has no preference about who to ask about whose capabilities of unblocking the blocked pathway (taking the Gricean Maxims as a basis, the statement by one agent about his own capabilities is as trustworthy as

about other agents' ones), she picks the first one from the list of equal valued question operations.

$$\Xi_{a_c} \leftarrow \Xi_{a_c} \cup (K_{a_c} \overleftarrow{\widehat{\Diamond}}^{\tau}_{(r_{has},(a_c,e_{sw}))}, \top, \iota(\tau)\begin{pmatrix}7 & 2\end{pmatrix}^T) \tag{80}$$

As a result, two interactions, both Bob–Alice and Carlos–Alice, are started by Bob and Carlos who have become aware of the sword that Alice carries. When Alice comes to know that Carlos does not know whether Bob can unblock the pathway or not, she ties in with the interaction started by Bob some moves ago.

```
ALICE uses the forest track departing from Amonlond.

                        BOB
                (to Alice)
        Please give me the sword!

                        CARLOS
                (to Alice)
        Please give me the sword!

                        ALICE
                (to Carlos)
        Well, can you unblock the
        river bridge way?

                        CARLOS
                (to Alice)
        No, I can't.

                        ALICE
                (to Carlos)
        Can Bob unblock the river
        bridge way?

                        CARLOS
                (to Alice)
        I don't know whether he can
        unblock the river bridge way
        or not.

                                                (CONTINUED)
```

```
CONTINUED:

                              ALICE
                      (to Bob)
              Can you unblock the river
              bridge way?

                              BOB
                      (to Alice)
              Yes, I can.

                              ALICE
                      (to Bob)
              Please do so!

                              BOB
                      (to Alice)
              Okay, I will do that, if you
              give me the sword.

 ALICE gives BOB the sword

 As a consequence, BOB unblocks the river bridge way.

 ALICE uses the river bridge way departing from Baernsari.
```

Figure 12: The game course with the modification of (80), when Carlos seeks the sword too.

In each of the courses in figures 10, 11 and 12, the game ends with Alice taking the (at that time unblocked) river bridge way arriving in Caradcalen, which corresponds her overall goal. The situation for all three courses is depicted by figure 13.

Varying other parts of the setup results in other game courses. Possessing the knowledge that the old dune trail connects to Caradcalen as well, there would be no need for Alice to trade her sword for the help of Bob. Other options include Carlos' hat or Bob's ring and the agents' knowledge and preferences related to them.

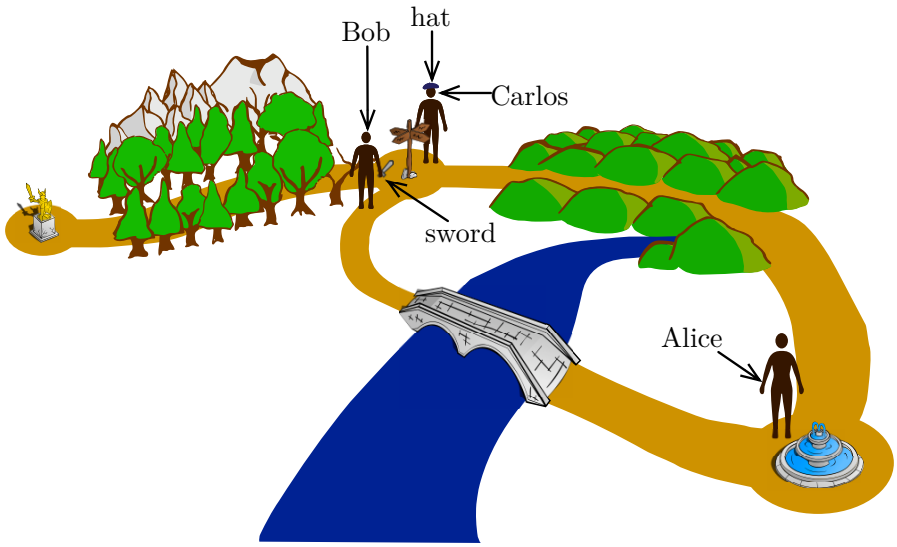

Figure 13: Alice has arrived at Caradcalen and thereby accomplished her mission.

5 Implementational details

In this chapter, data structures used by the implementation of the setup given in chapter 4 are depicted by UML class diagrams (see (Object Management Group 2007a,b) for the specification of the UML standard) to draw a line from the abstract specification of the given setup to a precise realization of the game principles.

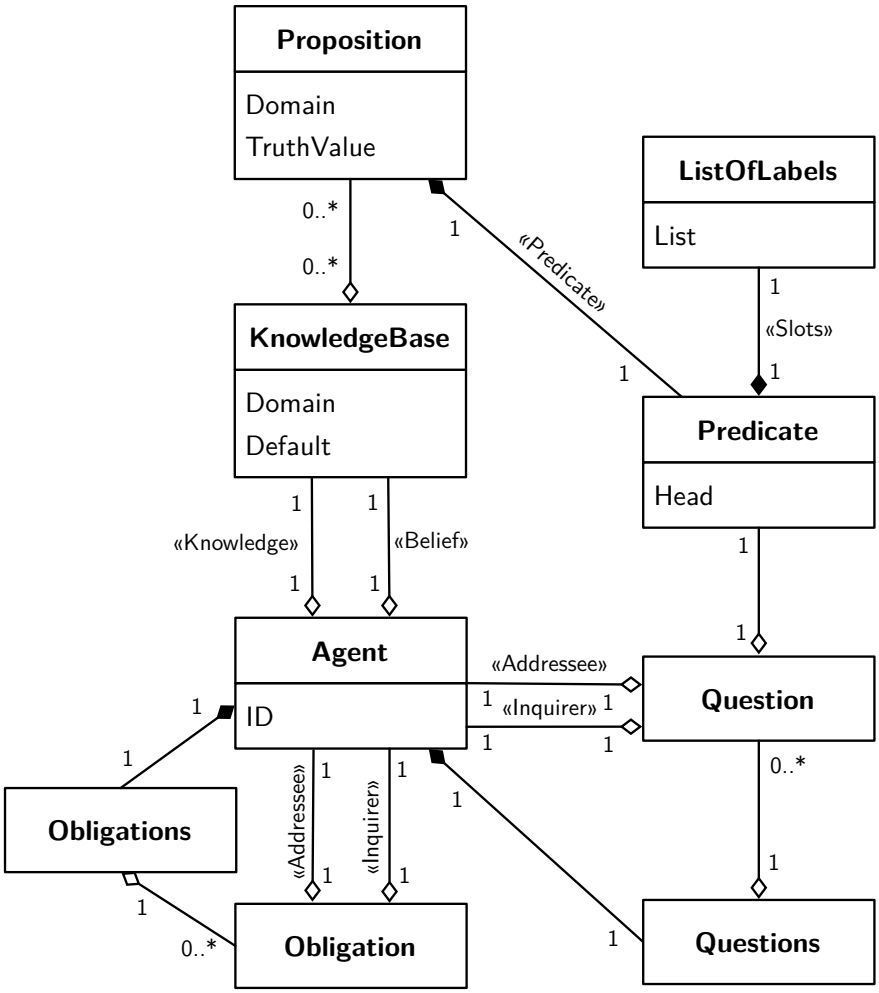

Figure 14: Class structure for all elements that aggregate to the *agent* class. The classes *KnowledgeBase* and *Agent* form in turn part of the class structure in figure 15.

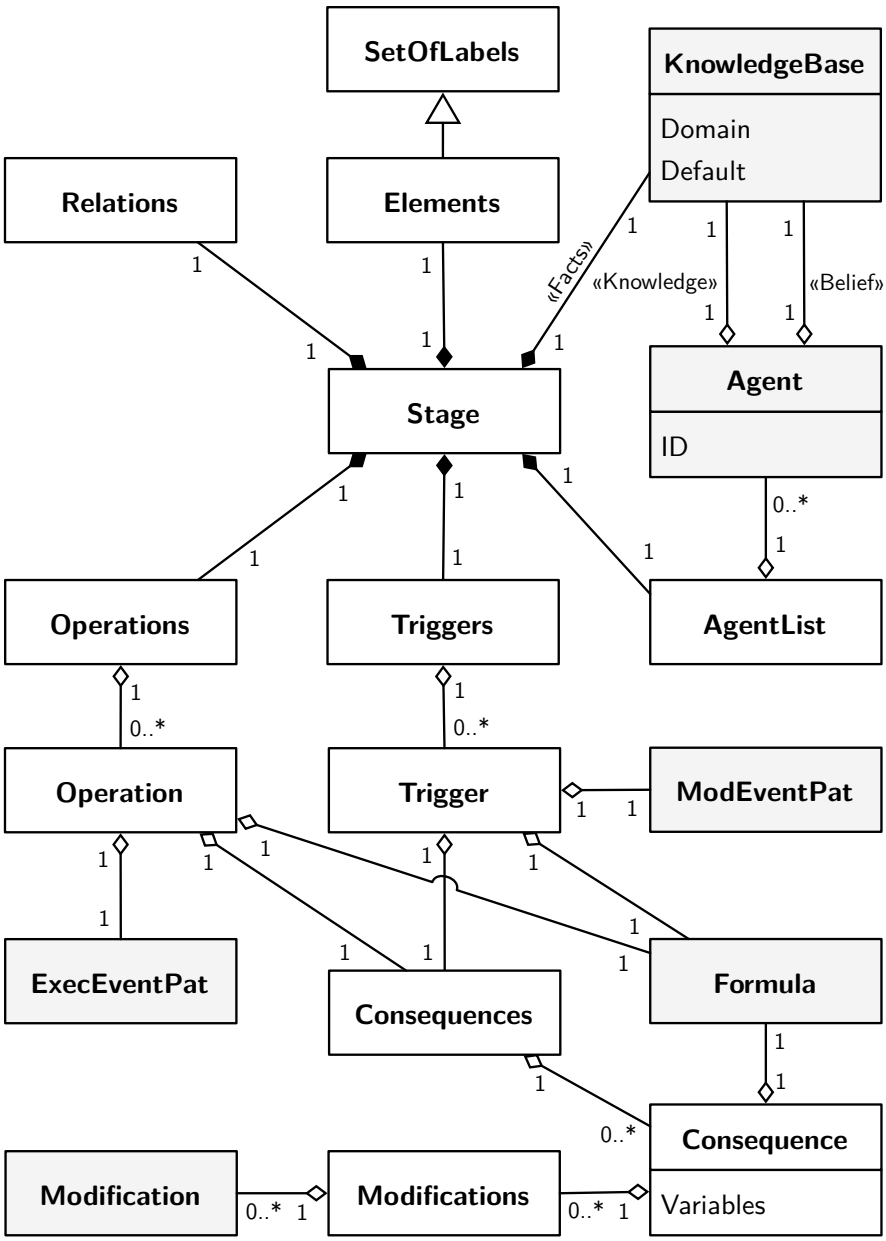

Figure 15: Main structure for the *stage*. Classes further specified elsewhere are indicated with a light gray background.

In figure 15, the composition of the stage is shown, while figure 14 goes into detail of the structure that is associated with the agent class.

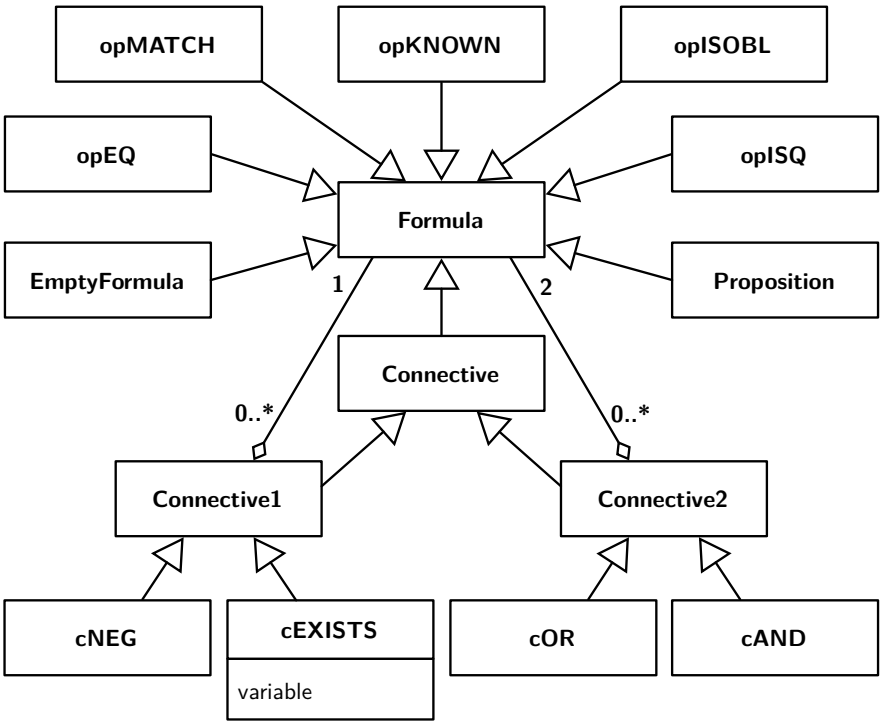

Figure 16: Structure of inheritance and association parting from the class *formula* which itself is associated to operations and triggers.

The inheritances of modifications and event patterns are depicted by the formulas 17 and 18, respectively.

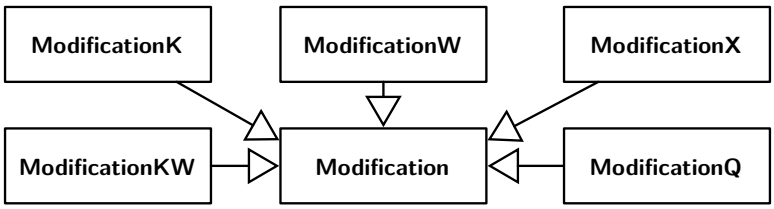

Figure 17: The inheritance structure of the modification class.

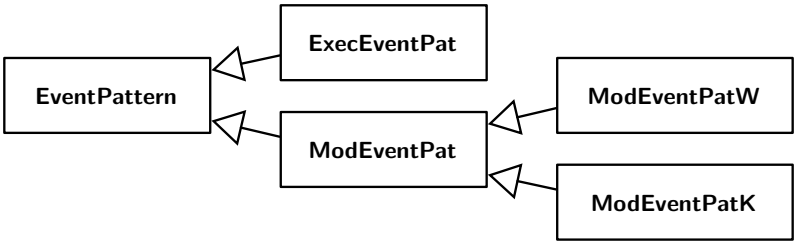

Figure 18: The inheritance structure of the event pattern class.

6 Concluding remarks

The aims of my work were achieved, but many ideas I outlined in this work remain to be kept up and implemented.

6.1 Conclusions

I defined a model for incremental generation of game moves where every move solely depends on the inner state of the participating agents at the time their decision–making counterparts select an actable action. Hence, the course of the game is based on decisions relying on the dynamics of the game world. Players can choose between all possible actions (i.e every combination of operations and parameters whose preconditions are fulfilled) which gives them the desired freedom of choice – within the limitations of the particular game world.

The concrete instantiation of this model I set up represents a small game with a limited amount of elements, operations and three agents, who are driven by simple personality profiles adapting the 'Big Five' personality traits of Costa and McCrae (1992).

No free–form input system has been implemented, since I spent all efforts on the implementation of the game itself. Natural language output is generated by a system of shallow rules that convert the symbols of actions and modifications into textual representations, which get coordinated and concatenated subsequently. Parameter agreement rules make sure that referring expressions are used and discourse markers are introduced wherever possible. All characters' actions are expressed the same way, that is to say the characters personality is not reflected its speech.

6.2 Future work

In order to enable human players to interact with the game directing one or more agent on the stage an input interface must be implemented. The obvious solution is to present the player a list of all possible actions – like artificial players are given these lists too, so that they can simply calculate which option fits best for them. The entries of this list could be described with natural language texts (either imperatively directed to the agent "Give the sword to Bob!" or descriptive "I give the sword to Bob."), but this ceases being feasible with the amount of operations and elements growing. Alternatively, solutions based on sentence–constructions by sequentially selecting the operation and then the first to the last parameter are supposable ("I give …" → "I give to Bob…" → "I give the sword to Bob.").

Notwithstanding, these kinds of selection–by–clicking–interfaces do not support free–form input either, but, rather, disclose a predetermined view of the game. For allowing the player to enter whole phrases, a method must be developed to map these phrases to actions, be it possible or impossible ones, and filter out references to out–of–world concepts or rather handle them in a somehow consistent way (without rejecting the input while returning an error message to the player or letting the game master intervene into a dialogue with an NPC for instance). Once an engine is running that is capable of mapping written input, the next consequential step would be to adopt a speech recognition system to make the player's efforts of typing become redundant.

Another important challenge to make the agents more realistic is to emphasize further their personalities. In my setup, different personalities are reflected in different behavior expressed by discriminative preferences for performing actions in the game world. What my approach does not cover is the influence of personality on language, i.e lexical, grammatical and structural decisions taken by the speaker. Modeling all aspects of impact on the personal language use would include the need to specify the past of every agent as well as the history of the world the game plays in, which is in any case exaggerated. Letting the personality influence the generated text in the context of a lexicon with correlated and categorized entries and a grammar allowing for variety in terms of complexity, among others, would be a good starting point, though.

On the part of the artificial players' intelligence, all means of artificial intelligence' planning are available. The actors could 'learn' how to 'play' their agents by reinforcement learning algorithms for instance. Equipping them with means to deduce from their agents' observations, they could set and change their goals during the game is played.

References

Adams, E. (2010). *Fundamentals of game design.* New Riders Pub (cit. on p. 4).

Austin, J.L. (1962). *How to do things with words.* Harvard University Press (cit. on pp. 9 sq., 32).

Bateman, C.M. (2006). *Game Writing: Narrative Skills for Videogames.* Charles River Media (cit. on p. 3).

Beun, R.J. (2001). "On the generation of coherent dialogue: A computational approach". In: *Pragmatics & Cognition* 9.1, pp. 37–68 (cit. on pp. 9 sq., 12).

Brusk, J. (2008). "Dialogue management for social game characters using statecharts". In: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology.* ACM, pp. 219–222 (cit. on pp. 11 sq.).

Brusk, J. and S. Björk (2009). "Gameplay design patterns for game dialogues". In: *Breaking New Ground: Innovation in Games, Play, Practice and Theory: Proceedings of the 2009 Digital Games Research Association Conference, London. Brunel University.* (Cit. on pp. 3, 6, 11 sq.).

Brusk, J. et al. (2007). "DEAL: dialogue management in SCXML for believable game characters". In: *Proceedings of the 2007 conference on Future Play.* ACM, pp. 137–144 (cit. on pp. 11 sq.).

Costa, P.T. and R.R. McCrae (1992). *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO-FFI).* Psychological Assessment Resources (cit. on pp. 20, 37, 55).

Crawford, C. (2003). "Interactive Storytelling". In: *The video game theory reader.* Ed. by M.J.P. Wolf and B. Perron. Routledge, New York, pp. 259–273 (cit. on pp. 4 sq., 13).

Cuayáhuitl, Heriberto (2009). "Hierarchical Reinforcement Learning for Spoken Dialogue Systems". PhD thesis. School of Informatics, University of Edinburgh (cit. on p. 10).

Frasca, G. (2003). "Simulation versus narrative". In: *The video game theory reader.* Ed. by M.J.P. Wolf and B. Perron. Routledge, New York, pp. 221–235 (cit. on p. 4).

Gee, J.P. (2007). *Good video games+ good learning: Collected essays on video games, learning, and literacy.* Vol. 27. Peter Lang Pub Inc (cit. on p. 4).

Ginsberg, M.L. (1990). "Bilattices and modal operators". In: *Journal of Logic and Computation* 1.1, pp. 41–69 (cit. on p. 9).

Gorniak, P. and D. Roy (2005). "Speaking with your sidekick: Understanding situated speech in computer role playing games". In: *Proceedings of Artificial Intelligence and Digital Entertainment* (cit. on p. 10).

Grice, H.P. (1975). "Logic and conversation". In: *Syntax and Semantics* 11, pp. 41–58 (cit. on pp. 10, 30, 34).

Harel, D. (1987). "Statecharts: A visual formalism for complex systems". In: *Science of computer programming* 8.3, pp. 231–274 (cit. on p. 11).

John, O.P. et al. (2008). "Paradigm shift to the integrative Big Five trait taxonomy". In: *Handbook of personality: Theory and research*, pp. 114–158 (cit. on p. 20).

Kacmarcik, G. (2005). "Question answering in role-playing games". In: *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05)*, pp. 51–55 (cit. on pp. 11 sq.).

— (2006). "Using natural language to manage NPC dialog". In: *Artificial Intelligence and Interactive Digital Entertainment* (cit. on p. 11).

Kerr, C. and D. Szafron (2009). "Supporting dialogue generation for story-based games". In: *Proc. 5th Artificial Intelligence for Interactive Digital Entertainment Conf*, pp. 154–160 (cit. on pp. 5, 12).

Koller, A. and R. Petrick (2011). "Experiences with planning for natural language generation". In: *Computational Intelligence* 27.1, pp. 23–40 (cit. on p. 10).

Koller, A. et al. (2010). "A scalable model of planning perlocutionary acts". In: *Proceedings of the 14th Workshop on the Semantics and Pragmatics of Dialogue* (cit. on p. 10).

Labrou, Y. and T. Finin (1998). "Semantics and conversations for an agent communication language". In: *Readings in Agents*, pp. 235–242 (cit. on p. 9).

Lebbink, H.J. et al. (2003). "A dialogue game to agree to disagree about inconsistent information". In: *7th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 83–90 (cit. on pp. 9, 12).

— (2004). "Dialogue games for inconsistent and biased information". In: *Electronic Notes in Theoretical Computer Science* 85.2, pp. 134–151 (cit. on pp. 9, 12).

— (2005). "A dialogue game to offer an agreement to disagree". In: *Programming Multi-Agent Systems*, pp. 199–223 (cit. on pp. 9, 12).

Lemon, O. (2011). "Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation". In: *Computer Speech & Language* 25.2, pp. 210–221 (cit. on p. 10).

Mairesse, F. and M.A. Walker (2010). "Towards personality-based user adaptation: psychologically informed stylistic language generation". In: *User Modeling and User-Adapted Interaction* 20.3, pp. 227–278 (cit. on p. 20).

Mateas, M. and A. Stern (2003). "Façade: An experiment in building a fully-realized interactive drama". In: *Game Developers Conference, Game Design track*, p. 82 (cit. on p. 11).

Mateas, M. and A. Stern (2004). "A Behavior Language: Joint action and behavioral idioms". In: *Life-like Characters. Tools, Affective Functions and Applications* 194, p. V202 (cit. on pp. 11 sq.).

McCrae, R.R. and P.T. Costa (1999). "A five-factor theory of personality". In: *Handbook of personality: Theory and research* 2, pp. 139–153 (cit. on p. 20).

— (2003). *Personality in adulthood: A five-factor theory perspective.* The Guilford Press (cit. on pp. 20, 40).

McQuiggan, S.W. and J.C. Lester (2006). "Learning empathy: A data-driven framework for modeling empathetic companion agents". In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.* ACM, pp. 961–968 (cit. on p. 11).

Merrick, K. and M.L. Maher (2006). "Motivated reinforcement learning for non-player characters in persistent computer game worlds". In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology.* ACM, p. 3 (cit. on p. 10).

Object Management Group (Nov. 2007a). *OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2.* Tech. rep. URL: `http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF` (cit. on p. 51).

— (Nov. 2007b). *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2.* Tech. rep. URL: `http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF` (cit. on p. 51).

Orkin, J. and D. Roy (2011). "Semi-automated dialogue act classification for situated social agents in games". In: *Agents for games and simulations II*, pp. 148–162 (cit. on p. 11).

Reckman, H. et al. (2010). "Learning meanings of words and constructions, grounded in a virtual game". In: *Semantic Approaches in Natural Language Processing*, p. 67 (cit. on p. 11).

Rieser, V. and O. Lemon (2008). "Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation". In: *Proceedings of ACL*, pp. 638–646 (cit. on p. 10).

— (2010). "Natural language generation as planning under uncertainty for spoken dialogue systems". In: *Empirical methods in natural language generation.* Springer-Verlag, pp. 105–120 (cit. on p. 10).

Searle, J.R. (1969). *Speech acts: An essay in the philosophy of language.* Cambridge Univ Press (cit. on pp. 9, 11, 30).

— (1976). "A classification of illocutionary acts". In: *Language in society* 5.01, pp. 1–23 (cit. on pp. 30 sq.).

Srivastava, S. (2010). "The Five-Factor Model Describes the Structure of Social Perceptions". In: *Psychological Inquiry* 21.1, pp. 69–75 (cit. on p. 20).

Stein, A. and J.A. Gulla (1997). "Inferring and Negotiating User Goals: Adaptive Dialogue Planning Using Abduction". In: *ABIS-97. Proceedings 5. GI-Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Saarbrücken. SFB 378*, pp. 127–137 (cit. on p. 9).

Walker, M. et al. (2007). "Individual and domain adaptation in sentence planning for dialogue". In: *Journal of Artificial Intelligence Research* 30.1, pp. 413–456 (cit. on p. 10).

Wolf, M.J.P. and B. Perron (2003). *The video game theory reader*. Routledge, New York (cit. on pp. 2 sq.).

Zukerman, I. and D. Litman (2001). "Natural language processing and user modeling: Synergies and limitations". In: *User Modeling and User-Adapted Interaction* 11.1, pp. 129–158 (cit. on p. 9).