Gameplay Design Patterns for Game Dialogues

Jenny Brusk

Dept. of Game Design, Gotland University
621 67 Visby, Sweden
jenny.brusk@hgo.se

Staffan Björk

Interaction Design Unit, Chalmers University of Technology & University of Gothenburg
412 96 Göteborg
staffan.bjork@chalmers.se

ABSTRACT

Dialogues are natural models for human communication and have also been used to model interaction within computer games. In this paper, we look at current models of dialogues from within the field of computational linguistics and explore their usefulness of games, and especially for the design of gameplay through interaction with non-playing characters in games. This is done by analyzing several examples of computer-based games and similar playful activities, both to see which models of dialogues are used but also to suggest possible ways of expanding gameplay through using other dialogue models. Uses of existing models for dialogues within games are identified but an additional model, the Game State-based approach to dialogues, is introduced. The possible implications of the changes in gameplay are described through the use of gameplay design patterns, offering a way to encode design knowledge explicitly and link that knowledge to other gameplay design pattern collections.

Keywords

Gameplay patterns, dialogue systems.

INTRODUCTION

Interacting with a game system, characters in a game world and the other players one is playing with can be said to be the core of the game activity. Performing actions that receive feedback that in turn provoke new actions can be described as having a dialogue, and this is the natural model for communication both between players and between game characters. This paper explores how novel gameplay can be created in games through the use of dialogues, based upon the techniques and models developed within computational linguistics.

In this paper we investigate how practical dialogue systems can inspire new forms of uses and implementations of game dialogue systems. This includes presenting a number of features typical for human dialogues that can be useful for creating conversational game characters.

The aim is to provide designers with a set of concept tools regarding how dialogues and gameplay can be combined in games. To this end we use *gameplay design patterns* [6], which express reusable gameplay design choices as semi-formalized and interconnected descriptions that may turn design possibilities into explicit options. Furthermore, the use of design patterns allows the discussion to be built upon close to 300 general patterns [6], as well as several smaller collections from related areas of character design for gameplay [16, 17, 18], thus linking the design options regarding dialogue to options regarding gameplay in general. Gameplay design patterns are marked by being in capitalized italics (and should be distinguishable from games by context).

The structure of this paper is as follows. Initially we will discuss dialogues per se and typical features present in dialogues. A classification of dialogue systems is then given to show the range of available technologies. After this, several games are analyzed from a perspective of how they implement dialogue systems and how this affects gameplay, with noticeable differences documented as potential gameplay design patterns. Based upon the deconstruction made in the analysis, new categories are created to describe the potential uses of dialogue systems in games from a gameplay perspective. Armed with this resource, and the identified gameplay design patterns, we discuss how the analyzed games could be modified through different uses of dialogue systems, showing how dialogue systems can offer new gameplay possibilities. The paper concludes with describing identified opportunities and challenges of implementing these possibilities.

BACKGROUND

A dialogue can be described as a mean for agents to exchange and coordinate information, build social cohesion, and achieve mutual understanding [2]. We interpret an utterance on basis of the context, or its use in a "language-game" [31], and by virtue of its conventional force [3], such as asking a question or making an offer. To regard utterances as performatives, that they do something rather than just describe the state of affairs, is the key point in speech act theory introduced by Austin [3].

Breaking New Ground: Innovation in Games, Play, Practice and Theory. Proceedings of DiGRA 2009

There are several ways in which natural language dialogue may come into play in games. Assuming the distinction between game (G), player (P), player character (PC) and non-player character (NPC), and stretching the notion of dialogue somewhat, we may distinguish between the following dialogue types:

- P-G: Games may be "voice controlled".
- P-PC: Player is directing his player character using dialogue.
- NPC-NPC: For commenting on the states and the events of a game.
- P-NPC: To provide players with background, quests, and social interaction.

One could also consider a PC-NPC combination, but this is not a main focus in the paper since it would also require a closer look at the role of player agency in games. Likewise, P-P dialogues are not discussed since designing them demand a natural focus closer to interaction design than gameplay design. This paper mainly focuses on dialogues between the player and NPCs (P-NPC), but other types of dialogues will be discussed to show other possible uses of dialogues.

Dialogue Control Elements

Speakers typically distribute the talk in a structured way to avoid overlaps and misunderstandings, and they may give feedback and correct errors during the interaction. Bunt [9] refers to these aspects as dialogue control functions, and several such concepts are potential features of games, i.e. from a game context it is possible to define them as gameplay design patterns. Thus, when these are presented here they are presented as if they were gameplay design patterns to enable their use in later sections. In cases where fitting gameplay design patterns already exist, we will refer to those.

Turn taking is a basic dialogue control element that concerns how we distribute and organize the moves in a conversation (see e.g. [23]). The gameplay design pattern *Turn Taking* [6] addresses the same issues for games but does not stress the communicative aspect of the interaction. Ordinary human-to-human conversations are processed through *Incremental Dialogue Processing,* where interrupts are possible due to the fact that the hearer can process and react to each contribution while it's being produced. The alternative can be referred to as *Chunk-based Dialogue Processing*, by which dialogue contributions (for instance utterances or phrases) are processed in chunks.

When a dialogue only can be initiated and/or controlled by one of the agents this is referred to as *Single-Initiative Dialogues*, while the alternative is labeled *Mixed Initiative Dialogues*. Regardless of which model is used, the agent currently in possession of the turn expects to get continuous *Basic Input Feedback* from the other participants as an assurance that they have heard and understood the message.

A hearer may be able to *Barge-In* while the speaker is talking, i.e. interrupt the ongoing exposition. This is not the same as feedback, since a barge-in causes the speaker to stop talking and give attention to the hearer, rather than the opposite. Using gameplay design patterns, one can describe systems supporting *Barge-In* actions as having speech acts as *Interruptible Actions* [6].

Types of Dialogue Systems

A system that can communicate with a user in natural language is usually called a dialogue system [15] and its commercial use has typically been to rationalize task-oriented human interactions, such as booking train-tickets or managing bank transactions.

Four main approaches for building dialogue systems have been classified ranging from the least complex to the most complex: Finite state-based, Form-based, Plan-based, and Agent-based [1].

Finite state-based systems can be modeled as a graph consisting of nodes, representing states, and directed, labeled arcs connecting the nodes denoting transitions between the states. At each state the dialogue system will generate a prompt, usually in the form of *Canned Text Responses* (for instance a prerecorded utterance) [15], requesting specific information from the user. The system will change state if it can match the user's input with any of the outgoing transitions from the current state. If not, it will either re-prompt or start over (for instance after a number of re-prompts). This means that the dialogue is both *Diegetically Consistent*, i.e. every exchange is within the scope of the domain, as well as *Context Dependent* since whether the input is accepted or not is dependent on the current state. Different from finite state-based systems, the *form-based* systems can fill several slots from the same input as well as handle input provided in different orders.

In *plan-based* systems, the system and the human user collaborate on a plan to solve a specific task in a stable environment, as in expert systems such as the *Circuit-fix-it shop system* [26]. In virtual worlds, however, the environment is dynamically changing which may change the preconditions for executing the plan. In these systems, the conversation requires a model of an agent with the ability to re-plan in order to achieve some goal or change goal if priorities change and are therefore referred to as *agent-based* systems (c.f. [10, 24, 28]).

EXAMPLES OF DIALOGUE SYSTEMS IN GAMES

To explore the possible uses of dialogue systems beyond the simple observation made above, a number of games are analyzed in the following sections. We have chosen commercial games and research projects that taken together cover a broad range of dialogue systems in use. The research projects *ELIZA* [30] and *Façade* [22] have been included because they are both publicly available as well as interesting from a game dialogue perspective. Furthermore, even though it is not presented as a game, *Façade* contains

a complete gameplay structure. Several of the examples are quite old, and there are two main reasons why they still have been chosen: First, they are among the first, or early, examples of games that use dialogue. Second, their relatively limited context and variety make them easier to analyze and describe than the more complex games that have come later.

ELIZA

Although not a game per se, the computer program *ELIZA* [30] allows a human agent to have a dialogue with a fictional Rogerian psychiatrist through a text-based interface (an example of P-NPC dialogue, or possibly P-G since the NPC is the only perceivable part of the game). Although it can be seen as a candidate to overcoming the Turing test (c.f. [25] for modern versions of this test), the experience is typically a playful one for users aware of the true nature of the psychiatrist consisting of trying to make the program say illogical or inappropriate utterances. The interaction with the system enforces a *Turn Taking* structure similar to *Turn-Based Games* [6]. After the first welcoming utterance from *ELIZA,* the program hands over the initiative to the user and waits for input, potentially for an infinite amount of time, hence making it a *Single Initiative Dialogue* system. Players have the possibility of *Free Text Communication* since they can type whatever they want to the system and the input is then handled through *Chunk-Based Dialogue Processing*. Further, since conversing is the only activity one can do in the system it is a trivial example of *Gameplay Integrated Conversation*, i.e. a game in which conversation is the main gameplay element and where the dialogue is an integral part of the game mechanics, if one considers the activity gameplay.

ELIZA never admits to not understanding input from users, and can maintain the illusion of always understanding players through the possibility of responding with context-free questions. This "error-handling" is represented to players as part of the conversation, and not as part of the interface, which can be described as the design pattern *Diegetically Consistent Dialogue*, i.e. all utterances are consistent with the represented environment the dialogue takes place within. The program does not try to parse the semantics of a user's input; it relies on transformation rules to create its output based on the user's input. This means also that ELIZA lacks a proper dialogue manager. Given that the whole experience of interacting with *ELIZA* is in the form of a dialogue, *ELIZA* can be said to make use of the pattern *Player Constructed World* [6], since all specific details of the conversation comes from the players. *ELIZA* can also be said to be able to manipulate a language-game, i.e. a small aspect of a language focusing on a specific activity and the actions related to the activity. Ordinary this would require a model of the context and the current state of the activity but given that Rogerian psychology consists of mirroring patients' statements as questions, *ELIZA* can avoid the need of having a detailed understanding of the overall development of the dialogue.

Zork

Zork [14] is an adventure game where the player takes on the role of an adventurer that explores a fantasy environment in the search for treasures. The player controls his avatar through text input and all responses are likewise in text, giving an example of P-PC dialogue in a game. Since nothing happens until input is given, the game is an example of a *Turn-Based Game* that uses *Chunk-Based Dialogue Processing*. Unlike *ELIZA*, the text given to the system does not represent a conversation with a fictional character but rather instructions for the system on what actions, mental or physical, the player wants the PC to perform. Although these instructions use the representation of the game, i.e. the game's diegesis, error and ambiguity handling require some of the dialogue to be on a meta-level. Thus, it does not have *Diegetically Consistent Dialogue*.

Given that *Zork* presents players with a diegetic game world, *Zork* cannot only rely on transformation rules as *ELIZA* to provide feedback. Instead it needs to relate the player's input to the current state of the game world. Specifically, this takes place on two levels: to determine if it makes sense diegetically (e.g. not trying to pick up a sword that is out of sight), and to generate an appropriate response. Some responses are simple uses of the gameplay design pattern *Canned Text Responses* while other require more or less complicated algorithms. In addition to these types of responses, *Zork* needs to inform players when it does not understand the input, and handle loading and saving, which would correspond to the P-G dialogue in our model.

The way *Zork* uses language interaction for manipulating game components makes it possible to consider utterances during gameplay as illocutionary acts [3]. This can be described as the gameplay design pattern *Illocutionary Interface*, since the interface to the game is through communicative acts regardless of whether the generated actions within the game are represented as dialogue or not. However, text can also be used to direct and communicate with other characters in the game, such as in the sequel to *Zork* where the player could tell the demon to kill the wizard by writing "`DEMON, KILL WIZARD`". In this case, the player could actually interact both with the game system (or the PC) as well as with other characters in the story through the same interface.

Allowing players free text input to the game parser means that players will probably have to experiment with what input is acceptable and not in the game. Since this interaction is indeed part of gameplay, *Zork* can be said to have the *Game Interface as Puzzle*, and this design can be said to have followed adventure games into the point-and-click versions of the genre.

Grim Fandango

Taking place in the Land of the Dead, the game *Grim Fandango* [19] makes use of the Aztec belief that dead souls wait for four years until they reach the ninth

underworld - the final home. The game is a single-player adventure game where the player controls Manny Calavera, who has to pay off the debts he gained from living a "less-than-perfect" life to be able to continue the journey himself. The gameplay takes place through making Manny move around, pick up and interact with objects in the environment, and start talking to characters he meets.

Unlike the previous examples, P–NPC dialogues in Grim Fandango are separated from other activities in the game. Talking to a NPC in the game takes the player into a different *Game Mode* [6] where the different phrases that Manny can say are listed in the form of *Canned Text Responses* that are consumed if selected. The selection of utterances depends on where in the narrative structure the player is, thus providing the basis for the pattern *Context-Dependent Dialogue*. Each phrase is furthermore coupled with a pre-recorded sound file that can be interrupted, which could be interpreted as having the pattern *Barge-In*, but as there are no effects of interrupting the disposition it can also be described as an interruptible *Cut Scene*. All dialogues are initiated and controlled by the player, thus making use of the *Single Initiative Dialogues* pattern.

Each dialogue is purposeful in some way, either as *Information Passing* [17], to provide *Diegetic Game Hints*, or even as a key to change the game state. The dialogues in *Grim Fandango* are also important vehicles for conveying the story and creating an atmosphere that is consistent with the stage set. The game characters are presented both implicitly, through for instance *Rumors*, and explicitly by talking to them in person. The dialogues can also be regarded as a way to socialize with the other characters, as some of the choices have less or no impact on game progression but more have the function of supplying the player with a *Freedom of Choice* [6].

The Elder Scrolls III: Morrowind

The Elder Scrolls III: Morrowind [4] (*Morrowind* henceforth) is a role-playing game played in first person perspective by a single player. The player starts by building the character, such as specifying its race, class, sex, skills and a number of additional attributes. The actions the player then chooses to perform will have impact on how the game progresses and how the player character is perceived by the other characters in the game. The game challenges, as in most RPGs, involve solving quests, exploring, fighting, joining guilds, trading and interacting with other characters in the game. A player can approach an NPC to initiate a P-NPC dialogue, and activating the dialogue interface freezes the game world, making it a different *Game Mode*. This gives consequences to game play such as being able to start a discussion with a guard while being chased by a monster (as long as the guard has not noticed the monster) and having no risk of being attacked until the discussion has ended. Similar to all the previous examples, players initiate and control the dialogue, making it a *Single Initiative*, *Turn-Taking* system.

Morrowind uses a dialogue system based on hypertexts connected to a database in which all interface objects containing dynamically changing text are stored, such that journal notes for instance are stored aside with dialogue content. Dialogue management is hence restricted to selecting the correct database entry for a specific hypertext keyword based on the current game state, location, and the NPC currently addressed, which spawns the new patterns *Location-Specific Dialogue, Character-Specific Dialogue* and a generated version of the *Context-Dependent Dialogue* pattern. The selection is also dependent on the PC–NPC relationship, hence introducing the sub-pattern *Relation-Dependent Dialogue* to *Context-Dependent Dialogue*. Although some of the phrases are diegetically social interaction, nearly all conversations with NPCs are functionally attempts to complete *Gain Information* [6] goals. Further, it is possible to use *Affective Communication* (Admire, Intimidate, Taunt and Bribe) to influence the NPC's disposition towards the PC and change preconditions for succeeding with a preferred action.

The Elder Scrolls IV: Oblivion

Like its predecessor, *Elder Scrolls IV: Oblivion* [5] (*Oblivion* henceforth) is a single-player computer role-playing game allowing players the freedom of controlling how their characters develop within a rich fantasy environment. Besides activities such as fighting, stealing and cast spells, players can interact with several hundred NPCs through a specialized interface for P-NPC dialogues accessible only when the relation between the PC and NPC is sufficiently good. In contrast to its predecessor *Morrowind*, dialogues in *Oblivion* are performed by selecting a phrase, hearing the response and choosing a new phrase in a turn-based manner until any of the participants ends it. The phrases are pre-scripted and the selection depends on location, current relation between the PC and NPC, and status of quests, i.e. the sequel also uses the patterns *Location-Specific Dialogue, Character-Specific Dialogue, Relation-Dependent Dialogue*, and *Context-Dependent Dialogue*. As for *Morrowind*, starting a dialogue puts the player in another *Game Mode* and the rest of the game world is paused until the dialogue is finished. In essence, this shows that the dialogue system is a separate system from the main game, both as the actions performed are different and as these actions do not take up time in the game world. The dialogue interface also provides access to another interface where players can try to improve (or worsen) the NPC's perception of the PC through a mini game that is focused upon recognizing facial expressions of the NPC.

Oblivion has already been analyzed for aspects of designing non-playing characters [16] as well as their social networks [17]. During this process several patterns related to dialogues were identified, including *Contextualized Conversational Responses, Free Text Communication, Gameplay Integrated Conversations, Ambiguous Responses, Awareness of Surroundings, Initiative,*

Emotional Attachment, Actions Have Social Consequences, Eavesdropping, Social Norm, and *Information Passing*, and to a lesser degree *Own Agenda, Sense of Self, Competing for Attention*, and *Either You are with Me or against Me*. Several of these have been mentioned in the earlier examples but for documentation purposes it should be noted that they were first identified in *Oblivion* although not studied in greater detail due to a different focus and space considerations. Due to space considerations, readers are referred to the previous work for details about these patterns although it should be mentioned that the *Eavesdropping* pattern is supported through NPC-NPC dialogues.

Façade

Façade [22] is an interactive drama where the PC is invited over to the married couple Grace and Trip for a social visit. Standing outside their door, the player involuntarily performs some *Eavesdropping* on the couple quarreling and it becomes obvious that they are having some serious marital problems. This opening serves as an introduction to the plot and depending on how the player acts from now on the story will take different turns. As for *Oblivion*, it has been a case study in an earlier paper [17], where the patterns *Emotional Attachment, Competing for Attention, Eavesdropping, Awareness of Surroundings*, and *Either You are with Me or You are against Me* were associated with it. The expressed design goal of *Façade* has been to create a drama in which agents interact socially with each other and with the player (i.e. both P-NPC and NPC-NPC dialogues), and where each action performed affects the behavior and attitude of the other agents [22]. The agents in *Façade* therefore show that *Actions Have Social Consequences*, i.e. their behavior and decisions depend upon the other participating agents. Grace and Trip start conversations with the PC to gain sympathy, showing examples of *Initiative* and *Own Agenda*. Objects in the environment trigger conversation throughout gameplay, showing that the NPCs have *Emotional Attachment* to them and that they have *Location-Specific Dialogue* as well as *Character-Specific Dialogue*.

In Façade the dialogue discourse actually constitute the major part of the story, and it seems reasonable to define a gameplay design pattern, *Dialogue-based Game Construction*, explaining this type of design decision applied on games. The goal can be described as unlocking a *Relation-Dependent Dialogue* between Grace and Trip, signifying that the player has changed the context to create the right *Context-Dependent Dialogues* (but it should be noted that players can set their own goals rather than the culturally implied one of helping the couple solve their marital issues).

As for *ELIZA* and *Zork*, players interact with the system through *Free Text Communication.* Although *Façade* only processes text after the return key has been pressed, i.e., by *Chunk-Based Dialogue Processing*, NPCs can initiate actions regardless of what the player is doing and the system therefore has *Mixed Initiative Dialogues.* That the time passes while dialogues take place and the people not involved in the discussion can perform other actions show that *Façade* also has *Gameplay Integrated Conversations*. Furthermore, *Façade* handles *Multi-Party Dialogues*, as all three characters can be engaged in the same dialogue. The size of the group talking is not completely in the player's control: one must consider that all characters can perform *Eavesdropping*. Further, characters can *Barge-In* on others and the NPCs show a *Sense of Self* in becoming irritated when interrupted.

Mass Effect

Mass Effect [11] is a space opera game that has received acclaim in popular press for having a novel dialogue system. Although having responses available due to the presence of character traits and players choosing type of response rather than exact phrasing have been present in earlier games, e.g. the *Fallout* series and *Morrowind* respectively, the quality of the writing and integration of the different parts may explain the positive reception. Another aspect given the positive response may in fact be related to the interface; dialogue options are organized in a pie menu with the same type of responses always appearing in the same place, which simplifies selection.

Mass Effect uses a *Single Initiative*, *Turn-Taking* system. Although the system can be compared to *Morrowind* or *Oblivion* since they both use the *Location-Specific Dialogue, Character-Specific Dialogue*, *Context-Dependent Dialogue*, and *Relation-Dependent Dialogue* to *Context-Dependent Dialogue* patterns, the experience is radically different. The reason is that the options presented in *Mass Effect* specify both what the PC will talk about as well as how, rather than the "database retrieval" style provided in *Morrowind*. Responses are also *Canned Text Responses* but support *Diegetically Consistent Dialogues* since the responses are recorded pieces of voice acting with lip-synched avatars.

The dialogue system is in practice a sub system of the *Mass Effect* game engine with only some information transfer to the overall game state. As such the dialogue system does not support patterns such as *Emotional Attachment, Competing for Attention, Awareness of Surroundings*, and *Actions Have Social Consequences*. However, the dialogues revolve around personal backgrounds and romances, which show that the mentioned patterns can occur on a narrative level rather than a gameplay level.

A Comparative Analysis

After having presented several examples of how dialogues are used in games separately, we can now compare them with each other and see how they relate to models of dialogues. By doing so the potential design space between the games can be identified and this provides the basis for the discussion later on how the games could be provided novel gameplay through design changes.

Grim Fandango, *Oblivion*, and *Mass Effect* use *Finite state-based* tree-structures, where each player choice unfolds the branch of the selected node. *ELIZA*, *Zork*, *Morrowind*, and *Façade* on the other hand do not easily fit any of the models used to explain dialogue systems. The reason why ELIZA does not fit is due to having a negligible internal state, for nearly all input the system remains in the same state as before. *Zork* uses a *Form-based*-like approach when trying to disambiguate users' input but does this for the current game state and not for the whole game. The dialogue engine in *Morrowind* resembles information retrieval systems that offer some kind of dialogue management, such as the *BirdQuest* system [12]. The agents in *Façade* are built using a *Plan-based* approach, but the method used differs from ordinary plan-based approaches in that it allows multiple agents (in this case Grace and Trip) to have joint goals and behaviors [21].

Although *ELIZA*, *Zork*, and *Façade* have aspects of existing approaches in their design, it might be more correct to state that they have a *Game State-based approach*. That is, even if *ELIZA* and *Façade* are not games per se, the state of the dialogue is represented through the complete game state. *Morrowind* fits this model partly as it uses a different game mode for dialogues but has some tree-based structures in it. However, the conditions determining what can be discussed in each conversation depend on the game state (primarily the NPCs' perception of the PC) and most dialogue can be retrieved in a random access manner.

THE DESIGN SPACE OF DIALOGUE SYSTEMS IN GAMES

By using the example games and identified patterns as starting point we now discuss how design choices can be transplanted or modified, and identify hypothetical (in the sense that they have not yet been found instantiated in a game) gameplay design patterns. As an initial observation, the examples show a range of possible uses for dialogues as interfaces to games. *ELIZA* and *Zork* use dialogues as the sole way to interact with the game and in the first case it is done completely within the diegesis of the game. *Grim Fandango*, *Morrowind*, *Oblivion*, and *Mass Effect* have dialogues as separate modes in the gameplay with no other activities occurring while the dialogue continues. *Façade* integrates the dialogue into the game interface and lets dialogue and other activities take place at the same time.

Allowing dialogues to take place simultaneously with other activities as *Façade* does, i.e. by using *Gameplay Integrated Conversations*, is a conceptually easy way to change designs. This can add stress and tension, e.g. having to convince town guards in *Morrowind* or *Oblivion* to let the PC enter the city gate while monsters are approaching. Besides making verbal expressiveness a potential game skill, this also can cause speed of typing to be important for players. It should be noted that this idea can be applied on *Façade* but at a higher level of detail; players can be challenged to provide NPCs with proper *Basic Input Feedback* by redesigning the system to support *Incremental Input Processing*, i.e. handling input per key strokes or tokens separated by blanks instead of per complete utterances. The romantic and intimidating aspects of dialogue in *Mass Effect* could likewise be expanded with requiring players to have appropriate body language and distance to achieve the desired effect. To integrate game dialogues with the overall gameplay, one has to equate communicative actions with other game actions as well as provide a support for these actions to be performed simultaneously. In *Façade* they used ABL (A Behavior Language) to accomplish this [21]. Another approach is to use statecharts [13] for modeling independent but synchronized behaviors, as well as integrating the dialogue state with the overall game state [7].

The personality of NPCs in *Grim Fandango* and *Oblivion* are expressed through dialogues, and in *Façade* and *Mass Effect* the character's emotional state can be perceived as well. These features can be further explored to also include social behavior, dependent on the characters' interpersonal relationship and the role they play in the situation in question. An example of how this can be solved using standard technology is presented in [8].

The examined games make use of dialogues for gameplay in different ways. In *ELIZA* the dialogue is the gameplay (in the sense that there is gameplay at all) while for *Zork* it is the interface that makes gameplay possible. *Grim Fandango*, *Morrowind*, *Oblivion*, and *Mass Effect* use dialogues to provide information to players about the game world and to progress the various plots. In one sense they all use dialogues as *Illocutionary Interfaces* since they can either change the game world or the progress of a narrative structure, but this may not be apparent to players before they make utterances. Taking an extreme view, one could finish any of these games without understanding the dialogues in the games. For *Zork* and *Façade* one would at least have to parse out the important words used, i.e. use *Game Interface as Puzzle*. Interacting with *ELIZA* without understanding the language the system responds in is unlikely to give a meaningful experience for any longer period of time.

Zork and *Façade* show that requiring players to perform actions that express an understanding of the dialogue is one way of integrating dialogue and gameplay. For *Zork* this consists mainly of figuring out what verbs, adjectives, and nouns can be used in the interface. In *Façade* it is manifested through the vagueness of the goal and the openness of how players can express themselves.

The examples listed above may be perceived as only applicable to natural language interactions, but scripted dialogues can also be improved by allowing a wider range of utterance options in specific situations. These options may be available on basis of for instance the dialogue history, the character's internal state as well as the interpersonal relationship. One example of a more complex system that uses a dialogue menu combined with *Canned*

Text Responses is the Augmented Conversation Engine [29]. In this system, the response from the NPC is selected from a matrix of trust combined with randomness, which summarizes to about twelve different possible answers to a specific user input.

The combat system in *Monkey Island* [20], having to know the right insults to verbally defeat one's opponent, provides another alternative. This can be generalized into the pattern, *Colloquial Mastery*, i.e. one has to learn the use of the language beyond simple information transferal so that one masters the idiosyncrasies of the current environment. For example, soldiers may need to begin and end every sentence with "sir" and outlaws may need to add curses to impress NPCs to fit the *Social Norm*. A technical more challenging option is to require players to use *Delicate Phrasing*, formulating utterances without revealing sensitive information or causing insults. In this fashion, game dialogues can challenge players' skills in expressing themselves– either as a standalone game or as part of the overall gameplay.

CONCLUDING REMARKS

In this paper we have explored how dialogue systems can be applied in games. Suggested uses have come from established techniques and concepts from computational linguistics as well as by identifying uses in some examples and generalizing them. The specific suggestions have been identified as gameplay design patterns although not full descriptions of these have been given. The patterns suggest several different ways in which dialogues can provide new gameplay challenges as well as how game interfaces can be designed.

Even though natural language interaction adds possibilities and advantages in the design of the user interface, it also introduces new problems and difficulties. The system may fail to create an interpretation or create an erroneous one, and repeated failures in understanding may cause frustration for the player. It is beyond the scope of this paper to in detail explore how to solve these issues, but possibilities include having a range of possible input modalities, letting the system choose reactions based on the global state (as e.g. *Façade* does by using its beat system), or trusting players tendency to try to find meaning in utterances (as *ELIZA* does). *Free Text Communication* gives players large degrees of freedom but it may still be important to make the range of reasonable options obvious to players. A possible solution to this is to let the system take initiative and help the player to learn how to interact. Another approach is to have an adaptive behavior, such that the NPC adjusts to the player's type of behavior.

Gameplay design patterns have been heavily used in this paper. Several new design patterns were identified through analyses of existing games. Further analyses of hypothetical re-designs have provided the basis for the new patterns including *Delicate Phrasing*, *Colloquial Mastery*, and *Incremental Input Processing*. These patterns have not been described to the level of detail in the original collection but by situating them in the context of dialogue systems and by grounding them in concrete game examples, the meaning and potential are hopefully evident.

Regarding the applicability of models from computational linguistics to games, the examples did not show perfect matches to existing categories. This may be due to specialization to the applications in question, but also due to the presence of characteristics typically not discussed in computational linguistic. An alternative approach, the *Game State-based approach*, has been introduced as has the observation that the intention of dialogue systems for games can differ from other dialogue systems in that cooperativeness is not always the intended design goal.

REFERENCES

1. Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L. & Stent, A.: Towards Conversational Human-Computer Interaction. AI Magazine (2001)

2. Allwood, J.: On Dialogue Cohesion. Gothenburg Papers in Theoretical Linguistics 65, Dept of Linguistics, University of Gothenburg (1992)

3. Austin, J. L.: How to do Things with Words, second ed. Harvard University Press, Cambridge, Massachusetts. (1962)

4. Bethesda Softworks LLC: The Elder Scrolls III: Morrowind (2003)

5. Bethesda Softworks LLC: The Elder Scrolls IV: Oblivion (2006)

6. Björk, S. & Holopainen, J.: Patterns in Game Design. Charles River Media, Hingham (2005)

7. Brusk, J. & Lager, T.: Developing Natural Language Enabled Games in SCXML. In Journal of Advanced Computational Intelligence and Intelligent Informatics - Special Issue on Intelligence Techniques in Computer Games and Simulations, vol. 12, No. 2, pp 156-163 (2008)

8. Brusk, J.: Dialogue Management for Social Game Characters Using Statecharts. In Proceedings of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2008). Yokohama, Japan, pp 219-222 (2008)

9. Bunt, H. 1994. Context and dialogue control. Think Quarterly 3, 19-31.

10. Cassell, J., Sullivan, J., Prevost, S. & Churchill, E. F. (Eds.): Embodied Conversational Agents. MIT Press (2000)

11. Demiurge Studios [Bioware]: Mass Effect (pc version) (2008).

12. Flycht-Eriksson, A., Jönsson, A., Merkel, M. & Sundblad, H.: Ontology-driven information-providing dialogue systems. Tampa, Florida, USA (2003)

13. Harel, D.: Statecharts: A Visual Formalism for Complex Systems, In Science of Computer Programming 8, North-Holland (1987)

14. Infocom: ZORK I: The Great Underground Empire (1980)

15. Jurafsky, D &, Martin, J. H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice

Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ 07458. (2000)

16. Lankoski, P. & Björk, S.: Gameplay design patterns for believable non-player characters. In: Situated Play: Proceedings of DiGRA 2007 Conference. (2007a)

17. Lankoski, P. & Björk, S.: Gameplay Design Patterns for Social Networks and Conflict. In: Proceedings of Game Design and Technology Workshop 2007 (2007b)

18. Lankoski, P. & Björk, S.: Character-Driven Game Design: Characters, Conflict, and Gameplay. Paper presentation at GDTW, Sixth International Conference in Game Design and Technology, 2008.

19. LukasArts: Grim Fandango (1998)

20. LukasArts: Monkey Island (1990)

21. Mateas, M. & Stern, A.: A Behavior Language: Joint Action and Behavioral Idioms. In Prendinger, H. and Ishizuka, M. (eds) Life-like Characters. Tools, Affective Functions and Applications, Springer (2004)

22. Mateas, M. & Stern, A.: Façade: An Experiment in Building a Fully-realized Interactive Drama. Game Developers Conference: Game Design Track. (2003)

23. Sacks, H., Schegloff, E. A., & Jefferson, G.: A Simplest Systematics for the Organization of Turn-taking for Conversation. Language 50, 4, 696{735. (1974)

24. Prendinger, H. & Ishizuka, M. (Eds): Life-Like Characters: Tools, Affective Functions, and Applications. Springer (2004)

25. Saygin, A. P., Cicekli, I., Akman, V.: Turing Test: 50 Years Later. Minds and Machines 10, 4 (November), 463{518. Kluwer Academic Publishers. (2000)

26. Smith, R.: Spoken Natural Language Dialog Systems: A Practical Approach. Oxford University Press (1995)

27. Splash Damage: Enemy Territory: Quake Wars. (2007)

28. Swartout, W., Gratch, J., Hill, R., Hovy, E., Marsella, S., Rickel, J. Traum, D.: Toward Virtual Humans. AI Magazine, 27, 1 (2006)

29. Swain, C.: The Augmented Conversation Engine – A System for Achieving Believable Conversation in Games and Interactive Stories. In Proceedings of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2008), ACM, pp 213-218 (2008)

30. Weizenbaum, J.: Eliza - a Computer Program for the Study of Natural Language Communication Between Man and Machine. Communications of the ACM Archive 9, 1 (January), 36-45 (1966)

31. Wittgenstein, L.: Philosophical Investigations. Blackwell Publishing (1953)