Evaluation of a Drama Manager Agent for an Interactive Story-based Game

Andrea Corradini¹, Manish Mehta² and Santiago Ontañón²,

¹ University of Southern Denmark, IFKI
6000 Kolding, Denmark,
andrea@sitkom.sdu.dk

² Georgia Institute of Technology, Cognitive Computing Lab
Atlanta 30332-0760, Georgia, USA
{mehtama1,santi}@cc.gatech.edu

Abstract. There has been a growing interest in employing drama management components in interactive fiction games. In this paper, we evaluate our drama management approach deployed in a re-implementation of the Anchorhead game. Twenty subjects, with different levels of expertise, took part in the study. We use players’ feedback as a basis for guiding the personalization of the interaction. The results indicate that our Drama Manager (DM) helps in providing a better play experience. Quantitative analysis indicates that the DM causes an average subjective improvement of 12.5% of the subjects’ play experience. A large value of the Pearson’s product-moment coefficient also indicates a strong correlation between less experienced adventure game players and higher ratings for the inclusion of hints during gameplay. Qualitative analysis from the user interview shows that all subjects noticed the difference in playing the game with and without the DM while the former way of playing was also the preferred one. Players found the hints helpful and funny even if frustrating when they cannot be exploited for proceeding in the game.

Key words: drama management, gameplay, evaluation, interactive fiction game

1 Introduction

There has been a growing interest in creating story-based interactive fiction games where the player is considered an active participant in the ongoing narratives. The component in charge of guiding the complete dramatic experience is called Drama Manager (DM) [2] or Director [5]. The DM employs a set of actions provided at appropriate points in the ongoing game whereby the player is guided toward certain aspects of the story. In previous work [8,9], we evaluated search based DM techniques in a simple text based implementation of the game Anchorhead [3].

We have developed a graphical version that incorporates the results from our previous work with many architectural enhancements [16] as well as tacking the issues of working in a real-time game, and on a game where the amount of actions the player can generate is unbounded due to the natural language interface. In this paper, we focus on evaluation of our new DM approach situated in the Anchorhead game.
We run an evaluation study that consisted in two different sessions. Quantitative analysis of the collected data indicates that the DM effectively improves the payers’ game subjective experience. In particular, a large value of the Pearson’s product-moment coefficient highlights a strong correlation between less experienced adventure game players and higher ratings for the inclusion of hints (generated by the DM) during gameplay. Preference for the inclusion of the DM can be evinced also from a qualitative analysis of user interview questionnaires. From such an examination, we could draw a series of lessons learned that can help further developing the strategies of the DM. Besides a preference to play with the DM, the large majority of players reported that hints are very helpful and also make the game more entertaining. Nonetheless, at times, overexposure to hints that cannot be exploited directly for proceeding through the different situations occurring in the game was considered frustrating.

The rest of the paper is organized as follows. In section 2 we present an overview of Anchorhead, the game used in our experiments, and our interactive drama architecture. We present an evaluation of our approach in Section 3 and 4. Section 5 outlines a few works related to our study. Finally, we conclude in Section 6.

2 An Architecture for Interactive Drama

In this section, we present both Anchorhead (the game used in our experiments) and our architecture for interactive drama that includes drama management.

Anchorhead is a text-based interactive story game created by Michael S. Gentry [3]. The complete game features a story divided into several days. In order to evaluate our architecture, we have implemented a subset of the story, consisting of day two, identified by [3] as interesting for evaluating drama management approaches. Graphical as well as text descriptions of the current scenario are presented to the player, who then enters commands in textual format, e.g. “enter the mansion” or “take the key”. Figure 1 shows a screenshot of our Anchorhead implementation.

In Anchorhead, the player explores a mysterious town since he has inherited a mansion there. By interacting with the people in the town, the player will uncover the hidden macabre secrets of the town of Anchorhead. The map is divided into 12 locations, and contains four non-player characters with which to interact.

Our architecture consists of six modules (shown in Fig. 2), namely:

- Graphical Interface (GUI): through which the user interacts with the system.
- Natural Language Understanding (NLU): parses the English text and generates a representation that can be understood by the game engine. The approach is based on our previous work [6].
- Game Engine (GE): responsible for running the game, maintaining the physical state, story state and a history of what happened during the game.
- Player Modeling (PM): develops a player model using case-based reasoning techniques from the player actions.
- Drama Manager (DM): takes the player model and the current game state and generates drama manager actions (DM actions) in order to influence the course of the game towards more interesting plots for the current player.
Chat Agent: when the NLU cannot understand the text written by the player and there is a character in the current location, the text is redirected to the chat agent, which handles conversations among player and characters.

A game is defined by specifying a map, a story, and a set of drama manager actions (or DM actions). The map defines the physical space of the game, it contains rooms, objects, characters, it specifies which items each character is holding, and the state of each object (if doors are open, closed, etc.). The story is specified as a set of plot points. A plot point is an important event in the game, for instance: “the player bribes the bum”. A plot point is defined as a set of preconditions, a trigger (typically a player action), and a set of effects (typically responses from characters in the game). Finally, the set of DM actions define the ways in which the drama manager can influence the game (as explained later). The set of actions the player can execute is defined by the sentences that the NLU module understands.

When a game starts, the graphical interface displays the current state of the game to the player and allows the player to enter textual commands (as shown in Fig. 1). When the player enters a command, it is sent to the NLU module. If parsing succeeds, the parsed action is sent to the game engine for execution. If the command is not understood, and there is some non-player character in the same location as the player,
the system assumes the player wanted to say something to that character. In that case, the command entered by the player is forwarded to the chat agent that attempts at generating an appropriate answer. For example, this happens when the player types commands like “How are you doing?”

Each time a player finishes a game, the system displays a form where the player can provide feedback. This feedback is then stored by the player modeling module in the form of a case. A case is a data structure that contains the trace of a game (the list of all the actions and game events that happened in the game) and the feedback that player provided. When a new player is playing the game, the player modeling module uses case-based reasoning (CBR) [1] in order to predict which aspects of the game the current player will like, by comparing it with previous players. The player model is used by the drama management module (DMM) in order to plan how to influence the game so that the satisfaction of the current player is maximized.

In order to do that, the DMM uses search-based artificial intelligence techniques to foresee the effects of different DM actions given all the possible actions a player can execute. See [8,9] for more details on how search-based techniques such as expectimax search can be sued for this purpose.

![Fig. 2. Main components of our interactive drama architecture.](image)

## 2.1 Drama Management in Anchorhead

The goal of a drama manager is to gently guide the player by providing hints or slightly changing the game, in order to maximize player satisfaction. For that purpose, the DM needs two things: 1) a way to guide the player, and 2) a way to evaluate player satisfaction. In our approach, the DM guide the player by executing DM actions. In particular our drama manager used a collection of 17 different DM actions, which the drama manager can chose to execute at any time in order to influence the story. Some of those actions are hints (e.g. “The Bum will hint that the crypt key is in the basement”), others are deniers (e.g. “The telescope will be out of order to that the player cannot look through it”), and others are causers (e.g. “The Bum will automatically bring up himself”).

In order to evaluate player satisfaction we used a combination of player modeling and predefined story aesthetic rules. In particular we used three aesthetic rules: thought-flow, activity-flow and manipulation (identified by Weyhrauch [12]), which measure that the story does not alternate too much among different sub-plots, that the
action does not alternate too much among different locations, and that the player does not feel too manipulated respectively. The DM selects then DM actions that maximize the evaluation of both the player model and the predefined story aesthetic rules. The expectation is that the resulting experience is more enjoyable to the player than when he interacts with the game without the DM.

3 Aspects of Cognitive Evaluation

Nowadays, many common software applications are taken for granted, even if most of them like Wikipedia, Second Life etc. are not even a decade old. These technologies have been increasingly affecting many aspects of our everyday lives and sometimes have even been considered as indispensable as traditional books. Not surprisingly, many researchers and scholars have thus focused their studies on the impact of these technologies and have introduced the ubiquitous concepts of usability and user experience as integral part of the broad human-computer interaction domain.

The analysis of these two concepts has become common practice for and has been applied to almost every study in HCI and closely related areas like computer games and digital entertainment. Nonetheless, there is still no single established theoretical and methodological approach and, in fact, there is a wide variety of methods some of which are radically opposed to one another but each of which has its own advantages and drawbacks. Some approaches tend to break down user experience into component elements in the attempt of discovering general models and rules. Others take contextual factors into account and employ more holistic and situated methods.

Perception pertains to the determination of the information that players seek out during game play. To that extent, researchers have typically tracked eye movements or measured search time and accuracy [13] to quantify visual search. We restricted our analysis of perception to the visual display of objects in the game that users interact with. Moreover, similarly to cognitive researchers who often use scaling methods to measure the strength of an event and/or object, we applied category scaling to measure system attributes like preferences, ease of learning, usefulness, etc. This was achieved with an automatically generated post-session questionnaire where questions were posed regarding the objects operated and/or selected.

To properly interact with the game, players must learn a substantial amount about the interface, such as the meaning of commands, the location of game objects, the contextual use of these items, the consequences of a command, and many more. While researchers often use more or less complex recognition tests and/or recall tests to study the processes involved in learning and memorization, we examined the recurrence of commands issued but not recognized by the system and looked at log files containing the interactive session to see whether players repeated specific errors. The same log files were also exploited to analyze users’ thinking in terms of strategy and problem-solving. With approximately half of the users, we performed preliminary process tracing by thinking aloud techniques and observation.
4 Methodology and Results

We carried out the study with twenty participants: eleven males and nine females, henceforth referred as P1 through P20. They were recruited among employees and students of a local University in Kolding, Denmark. The participants’ average age was 31.5 years. Subjects were of Danish, British, Italian, German, and Egyptian nationality, all with advanced skills in English. Forty percent of them (64% of males, 11% of females) judged themselves as experienced videogame players.

Each user evaluation consisted of 2 different sessions. The first session had an average duration of 60 minutes. At first, we collected and analyzed data about the user and his/her play style, preferences, previous gaming experience and favorite game genres. While user analysis has typically been restricted to demographics and general preferences, we took into account also the users’ general knowledge of the task domain and of the system to stress the importance of users' knowledge to their interactions with the game. Immediately thereafter, every player was provided with an explanation of the Anchorhead game. Following a five minute warm-up session the actual playing session lasting up to 25 minutes was started. Eventually, a post-test interview terminated the user testing. A second interactive session lasted an average of 40 minutes and consisted of another playing time of up to 25 minutes and with different settings to those employed in the first gaming session. At the end of the game time, one more post session interview was briefly carried out.

For the playing sessions, we uniformly split by gender the set of participants into 4 groups of 5 people. Henceforth, we refer to each of these groups as G1, G2, G3, and G4, respectively. Each group of people played the game under different conditions regarding the presence/absence of the DM as well as the use of previously collected games. Subjects were not told anything about the game settings. Procedurally, subjects played according to the following guidelines:

- Initially, subjects P1 to P5, belonging to group G1, played the game without the DM. We collected both the game log files and players’ feedback at the end of each game. We refer to the collected log files as CG1 game cases.
- Then, subjects P6 to P10, belonging to group G2, were first invited to play with the DM activated and utilizing the game cases CG1. Later they played without the DM. We collected both the game log files and players’ feedback at the end of each session. We refer to the collected log files as game cases CG2.
- Subjects P11 to P15, belonging to group G3, first played without the DM and then with it and utilizing the game cases CG1. We collected both the game log files and players’ feedback at the end of each game. We refer to the collected log files as game cases and indicate them as CG3.
- Later, subjects P1 to P5, belonging to group G1, were summoned again to play with DM activated and using all cases collected so far i.e. CG1, CG2 and CG3.
- Eventually, subjects P16 to P20 of group G4, played first with DM and using all cases CG1, CG2, and CG3 and later without DM.

The different game conditions account for: a) the presence/absence of the DM, and b) knowledge about previous players via the use of differently sized case libraries. During interaction, we observed and logged player’s actions, responses and reactions to the game. We analyzed the data obtained from both a numerical and a qualitative
perspective. From a quantitative analysis view, we were especially interested on the overall score that players assigned to the game both in presence and in absence of the DM. We also were interested in discovering whether and to what extent users prefer the addition of hints (thus ultimately the presence of the DM) during gameplay. The results are summarized in Fig. 3 where they are classified according to the four different subject groups along with the average over these groups. A value of 0 corresponds to the lowest rating, a value of 4 to the highest possible one. Figure 3 (left) clearly shows that players like the game more when they play with the DM enabled. The average score across all players for the game without DM was 2.0 while the average score with DM was 2.25 i.e. an improvement of 12.5% (Wilcoxon, p=0.021, Z=-1.95 one-tailed). The improvements over each group of subjects were: 8.3% for G1, 18.2% for G2, 0% for G3, and 25% for G4.

![Game Rating for Subject Group](image1)

![Hints Rating for Subject Group](image2)

Fig. 3. (left) Game ratings according to the different subject groups both with and without DM; (right) Average hints' ratings and their standard deviation values for each subject groups.

With the collected data we calculated the correlation coefficients to determine the strength and the direction of a linear relationship between several of the variables analyzed. As shown in Fig. 4, we calculated the Pearson’s product-moment correlation coefficient to relate the degree of experience of players with their game ratings, the degree of experience of players with their hints’ ratings, and eventually the game ratings with the hints’ rating. The Pearson’s coefficient is obtained by dividing the covariance of the two random variables being examined by the product of their standard deviations. Such a degree of linear dependence between two random variables tends to 1 (-1) in case of increasing (decreasing) linear relationships while it assumes some value in between in all other cases. The closer the absolute value of the coefficient is to 1, the stronger the correlation between the two variables. It must be noticed that a value of 0 for the correlation does not necessarily imply that the variables are independent since the Pearson’s moment coefficient detects only linear dependencies. The converse is however true i.e. independent variables are characterized by a coefficient 0.

As a consequence, our analysis indicates that people with less game experience tend to rate the game higher as well as to appreciate more game hints. According to our analysis, experienced players seem also to like hints but to a minor degree, which confirms previous studies we carried out [9]. Experienced players and high game rating are instead either not linear correlated or are independent. Pearson’s coefficient
statistics also reveals that subjects that like the game also like to receive hints as they are presented with the DM.

![Figure 4](image)

**Fig. 4.** Pearson's coefficient correlating player experience with hints and game rating.

In order to further understand the interaction, we performed a qualitative analysis using a simplistic version of a well-known qualitative analysis method, Grounded Theory [4, 10] as used in [11]. Grounded Theory is a research technique that operates almost in a reverse fashion to traditional research for it does not begin by researching and developing a hypothesis. Within such a framework, a variety of data collection methods under different conditions is employed as reported above. Instead of applying a specific model to the phenomenon under investigation, from the data gathered, we identified common patterns (the key points of Grounded Theory), which we then grouped into similar concepts. From these concepts we formed the categories mentioned in section 3 i.e. perception, memory, and language which form the basis for the creation of a reverse engineered hypothesis and which were analyzed singularly. Such a procedure allowed us to draw several main conclusions which we can now summarize in the following list of lessons learned.

**Lesson 1:** the DM techniques effectively increase player satisfaction. As depicted in Figure 3 (left), a numerical analysis of the interaction already permitted to reach the same conclusion. After playing first without DM, and later with the DM enabled, P5 said that “my problem is that I don’t play often but the second experience was much better than the first one”. During the interview following his second gaming experience, P4 asked if we made some change in the software since “it was much cooler to play this time than the last one”. From the questionnaire data, it emerged that all players perceived a difference between absence and presence of hints despite during their interaction episodes they were never told in which mode they were playing. They were actually not even told that there were different settings between their game episodes. In fact, P16 said that “The second time I played I was able to get more info out of the game; this was quite a good fun!”, assuming erroneously that it was her way of playing that caused the system to display more hints.

**Lesson 2:** the type and quantity of DM hints should depend on player’s experience. Again, the very same conclusion was already reached after numerical analysis of the interaction and highlighted in Fig. 3 (right). Subjects with limited experience with digital games (and especially of adventure games) who participated
to our study commented e.g. P10 said that “in the second session I got more hints and this was very helpful even if this did not help me much to finish the game” and P7 that “I wished more tips” and then added that “I was not always able to follow up the hints. For a beginner perspective, I need more of them to elaborate on what I can do. Maybe this does not apply to expert level gamers”. Moreover, less experienced players were more likely to be in trouble operating with objects and dealing with situations encountered in the game. P10 added that “I was not always sure what I could do in a given context. I would have wanted to be displayed a repertoire of commands instead of trial and error from my part”.

**Lesson 3:** the DM can effectively help in successfully playing the game. Several players pointed out and recognized the utility of the DM to solve riddles in the game. Conversely, players were frustrated when confronted with no hints since they had problems in advancing the game. After his first gaming episode without DM, P11 stated that “it was not clear how the game works, its objective, and how to accomplish it” and not surprisingly his game rating was very low. After playing a second time with hints enabled, his final score about the game increased and during interview he reported about the game that “once I got the hang of it, the tips made it easy for me to play”. Similarly, after playing first with the DM and then without it, during her second interview P16 reported that “I did not have the full overview of the game; maybe there should be more game possibilities and be more obvious”. This contradicts with one of her statements after her first episode when she said “I was pointed by the game to buy stuff in the magic shop but I could not get around it”.

**Lesson 4:** the content and quantity of DM hints should be tailored to the player’s experience. Currently, the strategy for the presentation of hints during gameplay does not take into account the player’s personal experience with the game. On that regard, more experienced users like P1 complained that “the system feedback and help messages are not always useful. Sometimes they are absurd or trivial”. This was an assertion which conflicts with comments of less experienced players as reported in the previous lessons learned item. Information about the player’s previous experience may thus help generating more appropriate hints. P17 suggested: “ideally you keep a model of all game objects and characters where you store their interaction history: in this way if I do something with an object or give it to somebody I don’t get the same messages or hints”.

**Lesson 5:** visual display of game entities must reflect the counterparts (if any) in the real world. This has also to be consistent throughout the game. In fact, a few users reported problems with the perceptions of some game objects. Sometimes, the 2D nature of the game with limited graphical implementation caused users to misrecognize game entities and ultimately to issue manipulative commands upon them that were not allowed. P17 commented that “some objects were recognizable by me some other I did not have a clue what they were. You cannot walk up to something if you cannot label them. The bed did not look like a bed”. Textual descriptions of the 2D graphical environment helped limit this kind of problem. Indeed, P17 continued saying that “I thought they were beds only because I knew I was in the bedroom and the command ‘examine’ confirmed I was right”.

**Lesson 6:** the overt representation of the text presented to display DM hints should reflect the grammar and the vocabulary that can be recognized by the system NLU. From a language point of view, all users quickly understood how to get around invalid
commands. Typically, after an average double retyping of a command that did not result in the expected game action, users tend to either try with a sentence reformulation with synonyms, or use the words occurring in hints and game descriptions or gave up. After playing for a few minutes, we saw a certain degree of convergence in the users’ use of commands. Players tend to make up and retrieve commands from a limited set of commands after finding out that they have worked in a precedent situation. Users just adapted these commands to the situation at hand. For instance, once a user realized that the command “look at <object>” causes the game to display a text and play a synthesized audio message with the description of that object, they rarely tried out later other possible commands to achieve objects’ examination. The speech capabilities of our system should however be increased to make the game more challenging for more demanding players. P3 stated that “the game was rather tolerant to my use of language”. On the contrary P6 commented that the “game does not always understand commands; these are limited and the grammar is too strict; it should be more flexible”. P11 said that “conversation is cool: the second time I played, the answers were better”.

Lesson 7: the type and quantity of DM hints should reflect a player’s strategy. The reason for such a conclusion is based on the observation that less experienced users do not follow any particular strategy. They tend to simply follow the hints provided by the DM, and thus remain unsure whether they could actually influence the evolution of the game. They also tend to enjoy more the interaction with non-playing characters since this kind of conversation is currently not focused on the solution of the game. Experienced gamers intelligently pick items that they feel would be useful later in the game and prefer not to waste much time with smalltalk with non-playing characters. As reported previously, most of the players (80%) admitted to have followed a strategy even if this was not always easy to notice. P17 said e.g. that “I developed a strategy along the way in the sense that when I realized that I can keep things, then I thought this is going to be useful to me”. A common strategy was to walk around and explore as many different locations as possible. As design insight for the future, we want to incorporate this observation in DM design. When the actions of a player are not doing anything logical to reach a sub-goal for some finite amount of time, the DM can provide a hint with the assumption that the player is confused. If the player was indeed trying to simply explore and did not like this hint at that game instance, he would provide a negative feedback for the particular hint. Later, players with similar playing patterns would not be given the same hint during these events.

Lesson 8: there should be a balance between game-oriented conversation and small talk when the player interacts with game characters. P20 reported that “I find the characters limited with regards to conversation-skills. I was hoping they would give me information about the Verlac-family but either they don’t know much about the family or they don’t want to talk about it”. P19 said that “I like to talk to characters. I tried to ask about specific questions but they always sneak out of conversation. So conversation was fruitless and frustrating after a while”. In a similar way, P8 found that “conversation with bartender is funny but if it stretches too long it becomes non sense”. Further, he added that “it was funny to talk to him. In general conversation was ok. But I always expected him to tell me about the game but he never did. The point of meeting people in the game must be to get info that you don’t get visually”.
5 Related Work

Bates [2] first proposed the idea of treating drama management as an optimization problem. The approach termed Search Based Drama Management (SBDM) was based on the fact that the drama manager chooses its best available action with expectation-calculating nodes. Weyhrauch [12] further developed the idea of a SBDM with a game tree based search that used an author specified evaluation function to measure the interestingness value for a particular story. However the DM employed was not connected to a concrete game and the techniques were tested using simulated players. Nelson et al. [7] define a Declarative Optimization based approach to Drama Management (DODM). The central premise of their technique is to provide the author with the ability to specify what constitutes a good story and use a reinforcement learning approach to optimize DM actions in response to player actions. This approach also uses a simulated player model to predict the next player action. Furthermore, the approach ignores a player preference model to measure the interestingness of a story from the player's perspective. In our approach to drama management, we construct a player preference model through real human player interaction with the game. Previous approaches employed only an author based evaluation function for story interestingness.

Façade [14] employs a beat-based drama management system suited towards tighter story structures where all the activity contributes towards the story. The Mimesis system [15] proposes a story planning based approach for real-time virtual worlds where the story plans are tagged with causal structure and the system handles player actions that might threaten the causal links through either re-planning the story or disallowing the player the opportunity to carry out actions. In such an approach though, only the author specifies the concrete goals that the planner should achieve; the approach doesn’t incorporate a player preference model. U-Director [19] is a decision theory-based approach to drama management which attempts to model goals and beliefs of the players. U-Director was only evaluated with synthetic players.

Finally, there have been several approaches evaluated with real players in the literature. PaSSAGE [17] is a system based on player modeling. PaSSAGE learns a model of the player's preferred style of play, and uses it to dynamically select the content of an interactive story. PaSSAGE was evaluated showing that in some groups of players (especially in females who found the game easy to understand) the system results in increased enjoyment. Another approach is that of Sullivan et al. [18] where they applied a DODM technique to a dungeon game called EMPath showing that players using drama manager are less lost in the game.

6 Conclusions

We have presented results from an evaluation study to measure the improvement in user experience with our drama management approach for the interactive fiction game Anchorhead. The results indicate that our DM causes an average 12.5% subjective improvement in the subjects’ play experience. They also show that less experienced adventure game players prefer to have hints provided to them during gameplay.
Qualitative analysis from the user interviews shows that all subjects noticed the difference in playing the game with and without the DM while the former way of playing was also the preferred one. The analysis further provided us with some improvements that need to be incorporated in the next system version. We plan to carry out experiments with more users to get more statistically significant results.

References