A Character-based Model for Interactive Storytelling in Games

Edirlei Soares de Lima
UNIDCOM/IADE
Universidade Europeia
Lisbon, Portugal
edirlei.lima@universidadeeuropeia.pt

Bruno Feijó
Department of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
bfeijo@inf.puc-rio.br

Antonio L. Furtado
Department of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
furtado@inf.puc-rio.br

Abstract **— In this paper, we present a novel character-based model for interactive storytelling in games, which combines multi-agent planning with a drama management strategy based on production rules to guide the narrative generation process. The underlying proposed method is capable of generating and controlling the enactment of character-based narratives in highly interactive game worlds. After describing the model, the paper proceeds to assess its applicability and performance. A fully implemented prototype game is used to demonstrate how the model works in practice.**

Keywords — interactive storytelling, games, narrative generation, drama management

I. Introduction

Interactive storytelling plays a fundamental role in games by allowing players to shape narratives with their own choice of actions. However, from the story writers' point of view, interactive narratives add a new layer of complexity to the development process. When writing a linear story, the author has full control over the narrative, but non-linear plots require different strategies to plan and represent stories. Most games handle non-linear narratives by representing the plot using branching structures with key choices presented to players at certain points, which allows writers to manually craft every possible storyline for the game. Although this approach has been successfully applied in many recent games, such as *Mass Effect* (BioWare, 2007-2012) and *The Witcher* (CD Projekt RED, 2007-2015), it imposes limitations to the story variety and can reduce the player's sense of agency [1][2].

Over the last decades, research on interactive storytelling has proposed several alternatives to hand-crafted branching structures, which can be categorized into plot-based, character-based, and hybrid approaches (detailed surveys can be found in [3][4]). Most previous works on the application of interactive storytelling in games follow plot-based approaches [5][6][7]. However, in some game genres, such as Role-Playing Games (RPGs), modeling the world as a simulation is more natural and offers some advantages over strict plot-based models, including the ability to support parallel storylines that can intersect with each other, increasing the number of possible stories and the possibilities for player interaction [8].

This paper presents a new character-based model for interactive storytelling in games, which includes a formal approach to specify the story world and a general architecture for the implementation of the model in games. The proposed method combines multi-agent planning with a drama management strategy to guide the narrative generation process. The main objective of the paper is to present our model and to validate its applicability and performance on a highly interactive game environment.

The paper is organized as follows. Section II reviews related work. Section III describes in detail the architecture of the proposed character-based model. Section IV explores the application of the proposed method in a game, with the help of a fully implemented prototype. Section V presents the results of a technical evaluation of the proposed method. Finally, concluding remarks are the object of Section VI.

II. Related Work

Most of the recent research on the application of interactive storytelling techniques in games has been focused on pure plot-based approaches. Breault et al. [5] propose a quest generation engine for games that uses a deterministic planning algorithm to generate linear quests that are consistent with a given world description and characters' preferences. A more dynamic approach is explored by Lima et al. [6][9], which combines planning, execution, and monitoring to handle nondeterministic events and alternative outcomes. In a more recent work, Lima et al. [7][10] present a quest generation method based on planning and genetic algorithms, which uses story arcs to guide the quest generation process according to a specific narrative structure. Their work was also extended to support the adaptation of existing quests according to user-specified story arcs [11]. Although these approaches can generate coherent narratives, the possibilities for player interactions with meaningful impacts in the plot are usually reduced to specific moments of the story (as normally happens with hand-crafted branching structures).

In character-based approaches to interactive storytelling, narratives emerge from the behaviors of autonomous agents that inhabit a virtual world. Although such models favor story diversity, it can also lead the plot to unexpected situations that are not complex enough to create an interesting drama. Therefore, most character-based interactive storytelling systems – including the one proposed in this paper – incorporate some form of drama management to guide the narrative generation or provide goals to the simulated agents. One of the earliest works to formalize the character-based approach for interactive storytelling is presented by Cavazza et al. [12], which uses Hierarchical Task Networks to represent characters' plans and to accommodate the authoring aspects of the baseline narrative. The use of authorial directives for the characters' specification is also often used to provide plot guidance to autonomous characters, such as proposed by Riedl and Stern [13] and Riedl and Young [14]. Another alternative involves the use of agent modeling tools to allow the author to define temporal causal relationships between scenes and characters' behaviors, such as proposed by Cai et al. [15]. Our approach to guide the narrative generation is similar to the use of behavioral directives proposed in [13]. However, instead of modeling characters'

978-1-6654-6156-6/22/$31.00 ©2022 IEEE

behaviors directly, we allow the author to establish production rules that are used to generate goals for the characters when certain situations are observed in the game world.

The techniques commonly used in planning-based narrative generation systems were also explored for other tasks in game contexts. Prominent examples include the Goal Oriented Action Planning (GOAP) [16] and the Mimesis architecture [17], which propose the use of automated planning to implement the artificial intelligence of non-player characters [18]. A recent work by Boeda [19] describes a multi-agent cooperation system for games based on GOAP and a messaging system, which allows non-player characters to communicate and cooperate with other agents to achieve goals. Although agents' actions and goals can be modeled to represent stories, the Boeda [19] system does not take into account any form of drama management. A similar approach is explored by Paul et al. [20], but using Hierarchical Task Networks to model the behaviors of non-player characters.

Recent research on character-based interactive storytelling has been focusing on different strategies to handle plot generation. Porteous and Lindsay [21] present non-cooperative multi-agent planning system for narrative generation where an antagonist obstructs the protagonist in recoverable ways, allowing the protagonist to eventually achieve his/her goal. As a general counter planning approach, their method combines plan generation, goal recognition, landmark identification, and re-planning. The automated interactions between protagonist and antagonist remove the need for authored narrative structuring information, but it is limited to narratives that revolve around the antagonist's obstructions to the protagonist's plans. In another research work, Porteous et al. [22] propose a narrative generation approach based on the concept of point of view, which uses a character's perspective on the story to create narrative variants. Their approach to control the content of the generated plots is based on the use of predicates that impose partial ordering constraints, which establish key components that are expected in the plot of well-formed narratives.

Although character-based models have been extensively explored in the area of interactive storytelling, there is a lack of research on the application of this approach in games, where narrative generation must take into account highly interactive worlds that can be affected by player actions.

III. Character-based Model for Games

A. Basic Definitions

We define the *narrative* of a game as a set of events that comprises actions performed by non-player characters (e.g., kidnaping, communicating, and giving items) and tasks to be accomplished by the player (e.g., fighting enemies, collecting items, and rescuing characters). While the actions of non-player characters are performed by the system, tasks associated with the player character must be carried out by the player. However, for the narrative generation process, both actions and tasks are treated equally; therefore, the term *action* is also used in this paper to refer to players' tasks.

The game is modeled as a simulation that comprises a set of characters $\{CH_1, CH_2, \ldots CH_n\}$. Each *character* is defined by a triple $CH_i = (N_i, G_i, P_i)$, where N_i is the character's name, G_i is a set of *goals* $\{g_1, g_2, \ldots, g_n\}$, and P_i is a sequence of actions $(a_1, a_2, \ldots, a_n)$ (called plan), which leads CH_i to achieve all goals of G_i. Both g_i and a_i are denoted by an atomic formula of the form $T(t_1, t_2, \ldots, t_n)$, where T defines the type of goal (e.g., *has*, *free*, *dead*) or action (e.g., *get*, *rescue*, *kill*) and the terms t_i (also called parameters) represent the elements involved in the goal or action (e.g., characters, locations, items). For example, *has*(*luke*, *master-sword*) represents a goal that indicates that *luke* (a character) must have the *master-sword* (an item). Similarly, *get*(*luke*, *master-sword*, *mystic-forest*) represents the action of *luke* getting the *master-sword* at *mystic-forest* (a location).

The game world is logically represented by a state (called *world state*), which consists of a set of positive literals (atoms) defining all characters, locations, items, and their current situation in the game world. An example of world state is:

character(*alice*), *character*(*crymson*), *location*(*white-castle*), *location*(*dark-tower*), *victim*(*alice*), *villain*(*crymson*), *free*(*alice*), *free*(*crymson*), *alive*(*alice*), *alive*(*crymson*), *at*(*alice*, *white-castle*), *at*(*crymson*, *dark-tower*), *protection*(*white-castle*, *3*), *protection*(*dark-tower*, 3), *relation*(*alice*, *crymson*, *negative*), *relation*(*crymson*, *alice*, *negative*),

which defines two characters (*alice* and *crymson*) and two locations (*white-castle* and *dark-tower*). While *alice* plays the role a victim, *crymson* acts as the villain. Both *alice* and *crymson* are free and alive. *Alice* is at the *white-castle* and *crymson* is at the *dark-tower*. The protection level of both *white-castle* and *dark-tower* is 3. The relation between *alice* and *crymson* is mutually *negative*.

An action is an instance of an *operator*, which defines all restrictions (e.g., spatial, temporal) for the occurrence of the action (preconditions) and the effects that result from the occurrence of the action in the world state (postconditions). More specifically, an operator is a triple $O_i = (ACT_i, PRE_i, POS_i)$, where ACT_i is an atomic formula with variables that defines the structure of the action O_i, PRE_i is a set of positive literals that establishes the preconditions for the occurrence of O_i,[1] and POS_i is a set of positive or negative literals that defines the effects of O_i. For example, the operator for the *kidnap* action is defined by:

O_1 = (
  ACT_1 = *kidnap*(*C1*, *C2*, *L*),
  PRE_1 = {*protagonist*(*C1*), *character*(*C1*), *character*(*C2*), *location*(*L*), *villain*(*C1*), *alive*(*C1*), *alive*(*C2*), *free*(*C2*), *at*(*C1*, *L*), *at*(*C2*, *L*), *relation*(*C1*, *C2*, *negative*), *know-unprotected*(*C1*, *C2*), *protection*(*L*) < 1},
  POS_1 = {¬*free*(*C2*), *kidnaped-by*(*C2*, *C1*)}
),

where *C1*, *C2* and *L* are variables, *negative* is a constant, and ¬ is a negation symbol. The literal *protagonist*(*C1*) is a special type of predicate used to identify the character that is expected to perform actions, which is dynamically updated according to the character that requests a new plan.

When a character CH_i has a goal ($G \in CH_i \neq \emptyset$), a planning algorithm is used to generate a sequence of actions $P \in CH_i$ to achieve the character's goal. The narrative plot is gradually

[1] We assume that preconditions of operators contain only positive literals to improve the computational performance of the planning process [23].

composed as the characters perform their actions in the game. As illustrated in Fig. 1, the dynamic nature of the simulation of multiple characters adds an extra dimension to the plot structure, where each character gains its own timeline of events. This dynamic structure also adds support to the occurrence of parallel events (two examples of possible parallel actions are illustrated in Fig. 1: a_2 in $P_1 \in CH_1$ and a_1 in $P_1 \in CH_3$; and a_1 in $P_1 \in CH_2$ and a_4 in $P_1 \in CH_3$).

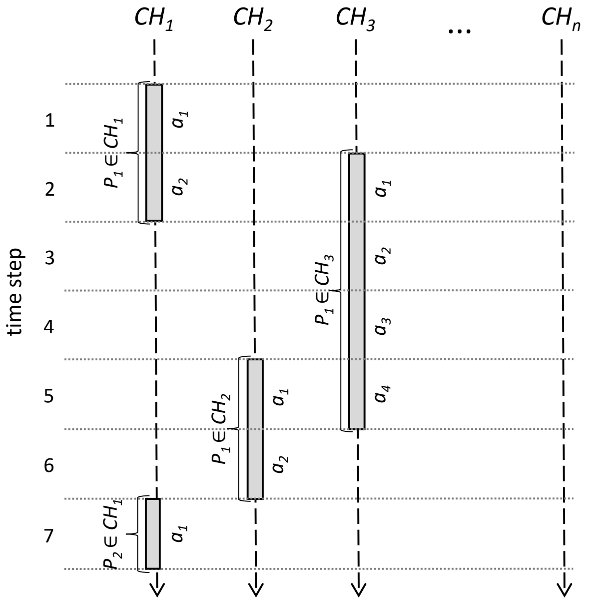

Fig. 1. The dynamic structure of the character-based plot.

Character's goals are generated by production rules, which are activated when certain situations are observed in the world state. A *production rule* is a pair $PRULE_i = (COND_i, GOAL_i)$, where $COND_i$ is a set of positive literals that defines the conditions (i.e., literals that must hold in the current world state) for the activation of $PRULE_i$, and $GOAL_i$ is a set of positive literals that establishes the goals that are generated by the activation of $PRULE_i$. Production rules are defined by the author of the story according to the expected behaviors of characters when certain situations occur in the game world. For example, the production rule establishing the behavior that leads a character to inform the villain that a victim is unprotected can be defined as:

PRULE$_1$ = (
COND$_1$ = {*protagonist(C1), victim(C2), villain(C3), at(C1, L), free(C2), at(C2, L), protection(L) < 1, relation(C1, C2, negative), relation(C1, C3, positive), relation(C3, C2, negative), alive(C3)*},
GOAL$_1$ = {*know-unprotected(C3, C2), at(C1, L)*}
).

The game world is also governed by general rules, which express knowledge about how the world works and evolves. A *general rule* is a pair $GRULE_i = (COND_i, EFFECT_i)$, where $COND_i$ is a set of positive literals that defines the conditions for the activation of $GRULE_i$, and $EFFECT_i$ is a set of positive or negative literals that establishes how the world state is affected by $GRULE_i$ (i.e., literals to be added or removed from the current world state). While production rules operate by generating goals for specific characters, general rules interfere directly in the world state. For example, a general rule to define that any character that is not at a protected location is unprotected can be described as:

GRULE$_1$ = (
COND$_1$ = {*at(C, L), protection(L) < 1*},
EFFECT$_1$ = {¬*protected(C)*}
).

B. The Architecture of the Character-based System

As illustrated in Fig. 2, the architecture of our character-based interactive storytelling system for games is composed of two main modules: (1) the *Character-based Simulation*, which controls the simulation of the character agents for the narrative generation; and (2) the *Game Manager*, which handles the execution of the characters' actions in the game while managing the actions that are requested to the player.

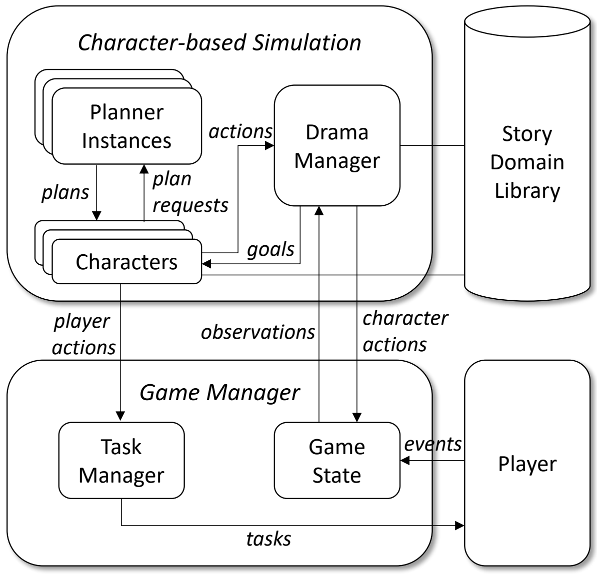

Fig. 2. Architecture of the proposed character-based interactive storytelling system for games.

As part of the *Character-based Simulation* module, a set of *Characters* represents the simulated agents, which are responsible for formulating their own plans of actions to achieve their goals using *Planner Instances*. In our implementation, we used the POPF planner of Coles et al. [24], which is based on a forward-chaining state-based search strategy and is compatible with our STRIPS formalism [25].

Characters' goals are generated by the *Drama Manager*, which is in charge of observing the world state for situations that can activate production rules and general rules. Although characters are responsible for planning their actions to achieve their goals independently, their actions are performed in the same game world. Therefore, changes in the world state produced by the actions of one character can cause inconsistencies in the plan of other characters. For example, if the plan of character CH_1 includes the action of getting an item on a specific location, but the item is collected by another character before CH_1 can reach the item's location, the plan of CH_1 will fail when executing the action *get* (the item will not be at the expected location). To guarantee the consistency of the plot, the *Drama Manager* always validates the characters' actions before executing them (the actions' preconditions are verified according to the current world state). When an inconsistency is identified, a replanning procedure is performed to find an alternative plan. If no alternative plan is found, the agent aborts the current goal.

The *Drama Manager* is also responsible for deciding when to allow characters to perform certain actions in parallel or selecting the order in which the actions occur. This decision is done according to the actions' relevance for the narrative (avoiding the simultaneous occurrence of important action types, which are identified by the author) or based on player choices (allowing the player to select how the events unfold).

Both *Drama Manager* and *Characters* have access to a *Story Domain Library*, which includes the definition of operators, the initial world state, production and general rules, and the expected narrative structure. In our implementation, the *Story Domain Library* is specified in an XML file.[2]

In the *Game Manager* module, the actions associated with the player character are transformed into game objectives by the *Task Manager*, which are presented to the player as the story progresses. While the player interacts with the game, his/her actions cause updates on the *Game State*, which is used by the *Drama Manager* to keep track of the current world state.

C. Plot Composition and Testing Tool

The process of authoring a character-based interactive narrative requires the specification of: (1) the initial world state, which includes the definition of all characters, locations, and objects that exist in the game world; (2) the operators that establish the possible actions that can be performed by characters; (3) production and general rules used to define the behaviors of virtual characters in different situations that can occur; and (4) the basic narrative structure that is expected to be present in the generated plot. These elements constitute the domain of the story, which we call *Story Domain Library*.

The general strategy to specify the domain for the story involves multiple testing phases to ensure that the production and general rules are enough to guide the autonomous characters towards meaningful situations. To assist authors in this testing process, we developed web-based plot composition tools, which allow users to visualize and test the actions of all characters in a timeline of events (Fig. 3).

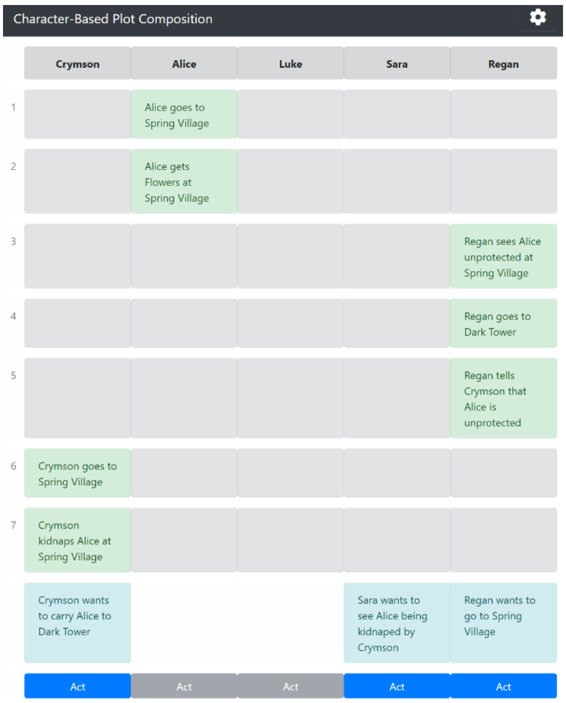

Fig. 3. Web user interface of the plot composition and testing tool.

As shown in Fig. 3, the plot composition tool presents the characters' actions in a table format. Each column is dedicated to a specific character, which is identified by the character's name displayed on the top of the screen (labels "Crymson", "Alice", "Luke", "Sara", and "Regan" in Fig. 3). In the bottom of the screen, buttons with the label "Act" are associated with each character/column. When a character has an action to perform in the simulation, the "Act" button is enabled (blue buttons) and a textual description of the character's intended action is presented above the button (light blue rectangles). When the user presses the "Act" button, the character's action is performed, and a textual description of the action is added to the character's column as a new row (light green rectangle).

An instance of the plot composition tool for experimenting with the story domain used in this paper is available at: http://www.icad.puc-rio.br/~logtell/character-based/. A video that shows how characters' actions affect the world state and how the author can use the plot composition tool is available at: https://www.youtube.com/watch?v=BGZtS1oF5C8.

IV. Application

In order to validate the applicability of our character-based model, we implemented a 3D RPG in the Unity game engine. The game incorporates the proposed architecture to dynamically generate and control the game's narrative, which tells the story of five main characters: the charming princess Alice, who lives under strict protection and knows the whereabouts of powerful items; a brave young hero called Luke, who is in love with princess Alice; the evil villain, Crymson, who is constantly waiting for a chance to kidnap Alice and become stronger; an evil merchant named Regan, who is ready to inform Crymson about anything that would be of his interest; and a friendly villager called Sara, who is ready to inform Luke whenever something bad happens to princess Alice. The game world is also inhabited by three support characters: Randall, Eric, and Abraham. Although support characters can provide items, challenges, and powers to main characters, they are not modeled as agents in the simulation to avoid unnecessary plan calculations for characters with limited actions. During the game, the player controls the brave hero Luke through several challenges to save princess Alice.

In the baseline story, Alice is kidnaped and imprisoned by Crymson, who forces her to tell him the locations of three elemental items that can provide Crymson with unstoppable powers. With this information, Crymson starts his search for the elemental items. Luke is informed about the kidnapping situation but knowing that he is not strong enough to defeat Crymson, he decides to depart on a journey to find a special sword that can increase his strength. After obtaining the sword, Luke confronts Crymson, but he is defeated by the evil powers of the villain. After recovering from the battle, Luke departs on a new journey to obtain a special shield that can help him stop Crymson's attacks. Luke obtains the shield by facing a knight's challenge. Protected by the shield, and with ample chance of success, Luke confronts and defeats the evil villain. Alice is released and the story ends with the marriage of Alice and Luke.

Several different stories can emerge from this basic storyline depending on how the simulation evolves (i.e., according to the order in which characters are allowed to perform their actions) and depending on the player decisions

[2] An example of domain library is available at: http://www.icad.puc-rio.br/~logtell/character-based/game-domain-library.xml

while performing the hero's tasks. Luke may or may not succeed on his first confront with Crymson depending on whether the villain can or cannot obtain the elemental power items before the battle. Luke can even be unable to defeat Crymson after obtaining the special shield if the villain has time to perform a transformation using the elemental power items, which will force Luke to try alternative ways to increase his strength. One of the alternatives involves receiving dark powers from the evil mage Abraham, which has a negative impact in Alice's feelings for Luke and may prevent them from getting married at the end.

The prototype fully implements the proposed architecture to generate and control the game's narrative, but it still lacks some gameplay elements commonly found in RPGs, such as non-story related enemies, combats, and level systems. Currently, the gameplay is driven by the tasks given to Luke (the player character), which include collecting items, travelling through different locations, and interacting with other characters. Cutscenes are used to present the actions of non-player characters. Fig. 4 shows a scene from the game prototype where Sara informs Luke that Alice has been kidnaped by Crymson.

Fig. 4. Scene from the game prototype: Sara tells Luke (the player character) that Alice has been kidnaped by Crymson.

V. Evaluation

Considering that one of the main drawbacks of automated planning algorithms is related to their high computational complexity, which grows according to the number of objects and operators involved in the planning problems, we decided to evaluate the scalability of the proposed character-based model through a performance test in different story domains with increasing levels of complexity. For this experiment, we created five variants of the story domain used in our prototype game (described in Section IV) with different numbers of characters, objects, operators, and facts in the initial state. Table I shows the main statistics of the story domains used in the experiment (*D3* represents the base story domain).

TABLE I. Statistics of the evaluated story domains.

	Story Domains				
	D1	***D2***	***D3***	***D4***	***D5***
Characters	3	4	5	6	7
Objects and Locations	20	25	30	35	40
Facts in Initial State	89	171	233	272	315
Operators	12	23	32	36	40

The performance test for each story domain was conducted using the proposed system to generate complete stories (random choices were performed when player interactions were required). Considering that during the simulation each character solves multiple planning problems with varying complexities (a new planning problem is solved every time a new goal is generated), we calculated the average, minimum, and maximum time spent by the planning algorithm to find solutions for the planning problems that occurred during the narrative generation. This process was repeated 10 times for each story domain and then the average time was calculated. The computer used to run the experiment was an Intel Core i7 7820HK, 2.9 GHZ CPU, 16 GB of RAM, NVIDIA GTX 1070 8GB GPU, using a single CPU core to process the planning algorithm. Fig. 5 shows the results of this test.

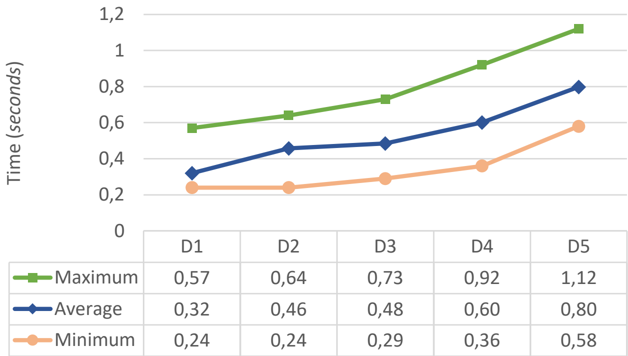

Fig. 5. Minimum, average, and maximum time to solve planning problems in five different story domains (time is presented in seconds).

The results of this experiment confirm the received wisdom that the computational performance of the narrative generation process is directed affected by the complexity of the story domain. By performing additional tests with variants of the story domains, we were also able to verify that the number of objects, locations, and facts in the initial state has no significant impact on the performance of the planner, which is mainly affected by number of operators and characters. Although this behavior affects the scalability of the system, we were able to obtain an encouraging level of variety and complexity with the story domain designed for our prototype game, which produces plots with approximately 50 events, 10 different storylines with major variations (i.e., plots with different endings or meaningful differences on how the events unfold), and uncountable storylines with minor variations (i.e., plots in which some events occur in different orders or plots with minor changes in content, such as the location where certain events take place).

We also conducted an experiment to evaluate the effects of the proposed narrative generation method in the real-time performance of our prototype game. For this test, we calculated the average, minimum, and maximum frame rates for three gameplay sessions of our game (approximately 15 minutes per session). The test was performed with the game running in a 1080p screen resolution (1920 x 1080) in the same computer used for the previous test. The results indicate an average frame rate of 134 FPS (frames per second), with a minimum of 89 FPS and a maximum of 163 FPS. Since the planner instances are executed in a process separated from the rendering task, no significant frame rate drops were observed when the planning algorithm was used to generate characters' actions. The observed variations in frame rate are related to the differences in the geometric complexity of various 3D locations of our game. These results indicate that our method can be applied in games without compromising their real-time performance.

VI. Concluding Remarks

In this paper we presented a new character-based model for interactive storytelling in games, which combines multi-agent planning with a drama management strategy based on production rules to guide the narrative generation process. The proposed model represents a viable alternative to branching structures and to strict plot-based approaches for narrative generation in games.

The results we obtained from the implementation of our model in a 3D RPG, along with the results of the performance test, offer a positive indication that our method can be effectively applied to highly interactive game environments. Although the developed prototype RPG is still very simple in terms of gameplay mechanics, it demonstrates the method's capacity to generate and control the execution of character-based narratives in real time. We expect that our model, enhanced by further research on narrative generation and user interaction methods, may contribute to the design of new forms of interactive storytelling in games.

As further research, we intend to conduct user studies to verify the adequacy of our model from the player's perspective. The implications of its adoption on the authoring process itself, both in terms of complexity and expressiveness, should also be determined. In this context, it should be clear that applying any schema-based method, like ours, to game development involves difficulties. The specification of the story domain library is a manual and time-consuming task, where developers must create a logical description of the game world, define planning operators, and establish production and general rules to guide the narrative. Therefore, finding ways to support the authoring process is an important future step for the complete validation of our model. In continuation to our project, we also plan to investigate the use of narrative structures, such as the *Hero's Journey* [26] and the *Grail Hero* [27], to more closely guide the narrative generation process.

Acknowledgment

We want to thank CNPq (National Council for Scientific and Technological Development) and FINEP (Funding Agency for Studies and Projects), which belong to the Ministry of Science, Technology, and Innovation of Brazil, for the financial support.

References

[1] R. Cardona-Rivera, J. Robertson, S. Ware, B. Harrison, D. Roberts, and R. Young, "Foreseeing Meaningful Choices," in *Proc. of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 10, 2021, pp. 9-15.

[2] S. Stang, "'This action will have consequences': Interactivity and player agency," *Game Studies*, vol. 19, no. 1, 2019.

[3] B. Kybartas, and R. Bidarra, "A Survey on Story Generation Techniques for Authoring Computational Narratives," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239-253, 2017, doi: 10.1109/TCIAIG.2016.2546063.

[4] A. I. Alhussain, and A. M. Azmi, "Automatic Story Generation: A Survey of Approaches," *ACM Computing Surveys*, vol. 54, no. 5, Article no. 103, 2022, doi: 10.1145/3453156.

[5] V. Breault, S. Ouellet, and J. Davies, "Let CONAN tell you a story: Procedural quest generation," *Entertainment Computing*, vol. 38, Article no. 100422, 2021, doi: 10.1016/j.entcom.2021.100422.

[6] E. S. Lima, B. Feijó, and A. L. Furtado, "Hierarchical Generation of Dynamic and Nondeterministic Quests in Games," in *Proc. of the International Conference on Advances in Computer Entertainment Technology*, 2014, Article no. 24, doi: 10.1145/2663806.2663833.

[7] E. S. Lima, B. Feijó, and A. L. Furtado, "Procedural Generation of Branching Quests for Games," *Entertainment Computing*, Article no. 100491, 2022, doi: 10.1016/j.entcom.2022.100491.

[8] M. Cavazza, F. Charles, and S. J. Mead, "Interacting with virtual characters in interactive storytelling," in *Proc. of the first international joint conference on Autonomous agents and multiagent systems*, 2002, pp. 318-325, doi: 10.1145/544741.544819.

[9] E. S. Lima, B. Feijó, and A. L. Furtado, "Player Behavior and Personality Modeling for Interactive Storytelling in Games," *Entertainment Computing*, vol. 28, pp. 32-48, Dec. 2018, doi: 10.1016/j.entcom.2018.08.003.

[10] E. S. Lima, B. Feijó, and A. L. Furtado, "Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning," in *Proc. of the XVIII Brazilian Symposium on Computer Games and Digital Entertainment*, 2019, pp. 495-504, doi: 10.1109/SBGames.2019.00028.

[11] E. S. Lima, B. Feijó, and A. L. Furtado, "Adaptive Branching Quests Based on Automated Planning and Story Arcs," in *Proc. of the XX Brazilian Symposium on Computer Games and Digital Entertainment*, 2021, pp. 9-18, doi: 10.1109/SBGames54170.2021.00012.

[12] M. Cavazza, F. Charles, and S. J. Mead, "Character-based interactive storytelling," *IEEE Intelligent Systems*, vol. 17, no. 4, pp. 17-24, 2002.

[13] M. O. Riedl, and A. Stern, "Believable Agents and Intelligent Story Adaptation for Interactive Storytelling," in Göbel, S., Malkewitz, R., Iurgel, I. (eds.) *Technologies for Interactive Digital Storytelling and Entertainment*, vol. 4326, 2006, pp. 1-12.

[14] M. O. Riedl, and R. M. Young, "Narrative Planning: Balancing Plot and Character," *Journal of Artificial Intelligence*, vol. 39, pp. 217-268, 2010, doi: 10.1613/jair.2989.

[15] Y. Cai, C. Miao, A. H. Tan, and Z. Shen, "A Hybrid of Plot-Based and Character-Based Interactive Storytelling," in *Technologies for E-Learning and Digital Entertainment*, vol. 4469, Heidelberg, Germany: Springer, 2007, pp. 260–273, doi: 10.1007/978-3-540-73011-8_27.

[16] J. Orkin, "Symbolic Representation of Game World State: Toward Real-Time Planning in Games," in *Proc. of the AAAI Workshop on Challenges in Game Artificial Intelligence*, 2004, pp. 26-30.

[17] R. M. Young, M. Riedl, M. Branly, A. Jhala, R. J. Martin, and C. J. Saretto, "An architecture for integrating plan-based behavior generation with interactive game environments," *Journal of Game Development* vol. 1, no. 1, pp. 51-70, 2004.

[18] J. Orkin, "Goal-Oriented Action Planning (GOAP)". https://alumni .media.mit.edu/~jorkin/goap.html, (accessed Abr. 20, 2022).

[19] G. Boeda, "Multi-Agent Cooperation in Games with Goal Oriented Action Planner: Use Case in WONDER Prototype Project," in *Proc. of the 17th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2021, pp. 204-207.

[20] R. Paul, D. Charles, M. McNeill, and D. McSherry, "MIST: An Interactive Storytelling System with Variable Character Behavior," in R. Aylett, M. Y. Lim, S. Louchart, P. Petta, and M. Riedl (eds.) *Interactive Storytelling*, vol. 6432, Heidelberg, Germany: Springer-Verlag, 2010, pp. 4-15, doi: 10.1007/978-3-642-16638-9_4.

[21] J. Porteous, and A. Lindsay, "Protagonist vs Antagonist PROVANT: Narrative Generation as Counter Planning," in *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019, pp. 1069-1077.

[22] J. Porteous, M. Cavazza, and F. Charles, "Narrative generation through characters' point of view," in *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 1297-1304.

[23] T. Bylander, "The computational complexity of propositional STRIPS planning," *Artificial Intelligence*, vol. 69, no. 1–2, pp. 165-204, 1994, doi: 10.1016/0004-3702(94)90081-7.

[24] A. J. Coles, A. I. Coles, M. Fox, D. Long, "Forward-Chaining Partial-Order Planning," in *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling*, pp. 42-49, 2010.

[25] R. E. Fikes, and N. J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, no. 3-4, pp. 189-208, 1971.

[26] J. Campbell, *The Hero with a Thousand Faces*. Princeton, USA: Princeton University Press, 1973.

[27] E. S. Lima, A. L. Furtado, B. Feijó, and M. A. Casanova, "Towards Reactive Failure-Recovery Gameplaying: The Fall and Rise of the Grail Hero," in *Proc. of the XV Brazilian Symposium on Computer Games and Digital Entertainment*, 2016, pp. 262–271.