A Pattern-Oriented AI-Powered Approach to Story Composition

Edirlei Soares de Lima[1(✉)], Margot M. E. Neggers[1], Marco A. Casanova[2], Bruno Feijó[2], and Antonio L. Furtado[2]

[1] Academy for AI, Games and Media, Breda University of Applied Sciences, Breda, The Netherlands
{soaresdelima.e,neggers.m}@buas.nl

[2] Department of Informatics, PUC-Rio, R. Marquês de São Vicente 225, Rio de Janeiro, Brazil
{casanova,bfeijo,furtado}@inf.puc-rio.br

Abstract. This paper proposes the use of narrative patterns as an effective guide to preserve thematic consistency in the composition of stories using Large Language Models (LLMs). Our approach drove inspiration from a well-accepted, thorough, and overarching classification of folklore types and the deservedly famous Monomyth characterization of heroic quests. The approach comprises both the pattern-guided composition of narratives and the creation of new patterns by applying the most specific generalization (MSG) criterion. We designed and implemented an AI-powered tool to create complex narratives in a storyboard format to demonstrate the feasibility of the proposed approach.

Keywords: Computational Narratology · Thematic Consistency · Story Composition · Narrative patterns · Most Specific Generalization · LLM

1 Introduction

The purpose of the present paper, situated in the *Computational Narratology* area [6,13], is to investigate how meaningful narratives can emerge under the guidance of noteworthy existing *patterns*, able to impose coherent narrative structures, and how to create new adequately expressive patterns by applying the notion of *most specific generalization* [15] to a number of texts.

We started by learning from one of the most influential literary researchers in narratology, Mieke Bal, that the study of narratives should encompass three distinct layers, which she respectively denominates [3]:

1. **Fabula** - a series of logically and chronologically related events that are caused or experienced by actors.
2. **Story** - a fabula that is presented in a certain manner.
3. **Text** - an artifact in which an agent tells a story in a particular medium, such as language, imagery, sound, buildings, or a combination thereof.

P. Figueroa et al. (Eds.): ICEC 2024, LNCS 15192, pp. 135–150, 2025.
https://doi.org/10.1007/978-3-031-74353-5_10

So, fabula constitutes the contents of the narrative, corresponding to *what* is told, whereas the story is *how* the fabula is told, and text is the *material form* through which the story is communicated. Mieke Bal explains that she is contemplating the readers (casual readers or learned critics), not the authors, and consequently, has in mind the reading and interpretation activity, rather than composition. However, for the purposes of automatic interactive narrative generation, it is convenient to start from the topmost fabula level, as we did in some of our previous works [9,10,20,24,32], specifying Propp-like functions [28] in terms of pre-conditions and post-conditions (effects), and building plots via backward-chaining plan-generation procedures [8,21,23].

With this approach, we were able to ensure logical coherence; however, by not contemplating Mieke Bal's story layer, the generated plots invariably looked shallow and, which is worse, often lacked *theme* consistency. In other words, the generated stories failed to meet the requirement that it should always be possible to summarize in a few words what a narrative is about.

In the folktale domain, this requirement is adequately contemplated through the characterization of *types* and *motifs* in the Aarne-Thompson-Uther comprehensive *Index* [34]. Being told that a tale is a variant of type ATU 333, *Little Red Riding Hood*, one could expect to read how "The wolf or other monster devours human beings until all of them are rescued alive from his belly" and, while reading, would observe how the type's conventional episodes are enacted:

1. **Wolf's Feast**. By masking as mother or grandmother the wolf deceives and devours a little girl whom he meets on his way to her grandmother's.
2. **Rescue**. The wolf is cut open and his victims are rescued alive; his belly is sewed full of stones and he drowns, or he jumps to his death.

Another well-known thematic guide is Joseph Campbell's *Monomyth* structure [4], which lists the episodes that should be found in any hero's journey, being effectively seconded by Christopher Vogler's *writer's journey* guide to prospective authors [35]. Motivated by the Campbell/Vogler proposal, we turned our attention to *narrative patterns* as a way to tackle interactive composition at a thematically consistent story level, adopting the Monomyth as a first example.

The remainder of the paper is organized as follows. Section 2 presents a short survey of related works. Section 3 describes our approach to use narrative patterns. Implementation details are described in Sect. 4. Section 5 contains conclusions and future work directions.

2 Related Work

The use of LLMs for narrative generation is a recent research topic. While some works focus on the use of fine-tuning techniques to improve narrative quality [1,12,36], others explore new strategies to handle the narrative generation process [5,37,38,40,41]. There are also works that focus on the evaluation of the writing capabilities of LLMs [7,42,43].

Our work represents a new strategy to explore the storytelling capabilities of LLMs. In this context, Wang et al. [37] use interpolation techniques to guide LLMs in the process of producing coherent narratives with user-specified endings. Castricato et al. [5] propose to treat story generation as a question-answering process. Yang et al. [41] explore the use of an LLM to generate a structured plan for the story, and then use recursive reprompting to incorporate relevant information into the generated story. Xie and Riedl [39] employ LLMs to generate suspenseful stories using an iterative-prompting-based planning approach. Lima et al. [18,19,25] use semiotic relations to allow an LLM to create new narratives by reusing elements from existing stories.

Extending our analysis beyond LLMs, we can identify other strategies to use narrative patterns for story composition. Alvarez and Font [2] introduced a system for representing narrative structures using interconnected tropes [16] extracted from TvTropes [30] and graph grammars to describe game narrative structures at an abstract level. Lima et al. [22] propose a method that reuses existing stories to generate new narrative variants by combining episodes extracted and adapted from different stories that share the same narrative structure.

Although the use of LLMs for narrative generation is a very active research area, to the best of our knowledge, this is the first work to propose the use of LLMs for pattern-guided narrative composition and creation of new patterns.

3 Narrative Patterns

3.1 Pattern-Guided Composition

Starting from the highly influential Campbell/Vogler Monomyth, which constitutes the initial motivation of the present work, we considered its fundamental list of twelve stages and proceeded to adapt it to our standard pattern format, which coincides with how the Aarne-Thompson-Uther Index formulates the episodes of type ATU 333 [11,34]. Therefore, each episode was specified by a *stage name* and an expressive *stage description*. By doing that to the Monomyth list, with the help of GPT 4, we obtained what we named the Hero's Journey pattern, which is described at https://narrativelab.org/patternteller/pattern.php?id=6.

To conceive the Monomyth, Campbell has surely analyzed, looking for their commonalities, a considerable number of heroic narratives. And in the analysis was famously influenced by Carl Jung's archetypes. For instance, in the place of Propp's 'villain', he included the 'shadow', who in Jungian interpretation [17] "symbolizes our 'other aspect', our 'dark brother', who, albeit invisibly, yet belongs inseparably to our totality."

Another well-known structure was proposed by Blake Snyder in a book [33] which, as announced in its title - *Save the Cat: the Last on Screenwriting You'll Ever Need* - reaches the utmost level of generality, not being confined to heroic genres. In the opposite direction one can cite Lord Raglan's heroic structure [29], less generic than the Monomyth, which, in particular, matches King Arthur's legend, including contradictory assertions to the effect that that"His body is not buried, but nevertheless he has one or more holy sepulchers". Even more specialized is our pattern *The Fall and Rise of the Grail Hero*, which accompanies the evolution of the Arthurian hero along this supremely spiritual quest [27].

In our experiments, we have mainly used Vogler/Campbell's *Hero's Journey* and our *The Fall and Rise of the Grail Hero* pattern. Particularly engaging was the result of trying the *Hero's Journey*, with the premise "The young samurai Yoshitsune, living in the twelfth century, recovers the sacred mirror of Amaterasu which had been stolen by Taira no Tomomori". The resulting story is available at: https://narrativelab.org/patternteller/view.php?id=246.

3.2 Creating New Patterns

New patterns can be created from what there is in common in two or more narratives. An adequate method to achieve this purpose is *most specific generalization* (MSG), as proposed in [15]. MSG can be easily implemented in logic programming and works well with symbolic logic expressions. However, its application to natural language narratives was beyond the state of the art at the time of the work mentioned above.

Fortunately, the difficulty has been reduced with the advent of LLMs, as we have found quite early through a simple example submitted to GPT 3.5 (gpt-3.5-turbo-0125): "Please apply the most specific generalization to the sentences: 'Lancelot loved Guinevere' and 'Tristan had a passion for Isolde', trying to capture the maximum in terms of commonalities." The first response was "A man felt deep affection for a woman." We then insisted: "Noting that Lancelot and Tristan were knights and Guinevere and Isolde were queens, could you achieve a better solution?", to which the system's reply is presented in Output 1.

Output 1. *Certainly, taking into account the additional context that Lancelot and Tristan were knights and Guinevere and Isolde were queens, a more refined most specific generalization could be: "A knight adored a queen passionately". This generalization maintains the romantic aspect while also incorporating the roles of the characters involved. It captures the essence of both sentences while highlighting the noble status and the intensity of affection between the characters.*

We thus felt more confident to adopt MSG for pattern creation, aware that, although the outcome might later need to be refined, a moderately satisfactory version would be reachable in most cases, without requiring extensive editing.

As a first experiment, we asked the prototype - which will be presented in Sect. 4 - to generate what we named the *Magnificent Protectors* pattern from two films whose similarity was evident, namely *Seven Samurai* (Toho, 1954) and *The Magnificent Seven* (United Artists, 1960), obtaining the pattern presented at https://narrativelab.org/patternteller/pattern.php?id=8. Adding it to a predefined pattern repertoire would be justifiable if it might be shown to have an ampler scope than that of the input narratives. In the case of the *Magnificent Protectors* pattern, GPT 3.5 gave some support to our decision to keep it, first by agreeing that the film *Avatar* would also *match* it, and next, called to *search* for yet another matching film, it cited *The Lord of the Rings: The Two Towers*. A welcome, unexpected sub-product of these tests was that, in both cases, the GPT 3.5 response took the *stage name* - *stage description* standard pattern format, as it revealed one-by-one the event correspondences.

Our prototype can handle more than two narratives and can even treat a single one, making it somehow more general. The input, in all cases, can be either the narrative title or a textual summary. We also found that the input can be a list of operations in clausal format, denoting a plan-generated plot: "travel(Jane, forest), capture(Draco, Jane, cavern), warn(King, Brian), travel(Brian,cavern), rescue(Brian,Jane)", from which the prototype created the pattern:

1. **Departure** - The protagonist begins a journey into an unknown and potentially dangerous environment.
2. **Encounter with Danger** - The antagonist captures the protagonist in a hostile environment.
3. **Alerting of the Saviour** - An authoritative figure warns a potential savior about the protagonist's situation.
4. **Saviour's Journey** - The savior embarks on a journey toward the location of the captured protagonist.
5. **Rescue and Resolution** - The savior rescues the protagonist from the antagonist's capture.

In a complementary open-access paper [26], we argued, based on seminal studies [14,31], for the existence of five fundamental genres: comedy, romance, tragedy, satire, and mystery. We then built patterns for each genre and applied these patterns to create genre-oriented stories starting from the same premise.

4 PatternTeller - An AI-Powered Storyteller

The proposed prototype, called PatternTeller, explores the structural knowledge present in narrative patterns to guide LLMs in the process of generating narratives. Unlike purely LLM-driven approaches, PatternTeller's structural integration aims to produce narratives that exhibit both the creativity of LLMs and the structural logic and complexity of human-authored stories. The system is accessible through our public website: https://narrativelab.org/patternteller/.

4.1 System Architecture

The architecture of PatternTeller, as shown in Fig. 1, is based on the multi-AI-agents approach for narrative generation proposed in our previous work [18,19]. Three distinct AI agents are employed: (1) a Storywriter AI Agent, which operates in the narrative generation process by producing story chapters composed of story events and scene descriptions; (2) an Illustrator AI Agent, which utilizes the textual scene descriptions to produce visual representations for the story events; and (3) a Pattern Specialist AI Agent, which is responsible for extracting narrative patterns from existing stories to compose new narratives. Coordinating these agents is the Plot Manager, who controls the overall narrative generation process by requesting chapters, patterns, and illustrations from each respective agent. User interaction is facilitated through the User Interface component, which allows users to compose and visualize narratives in a storyboard format. Stories and narrative patterns generated by the system are stored in a Database.

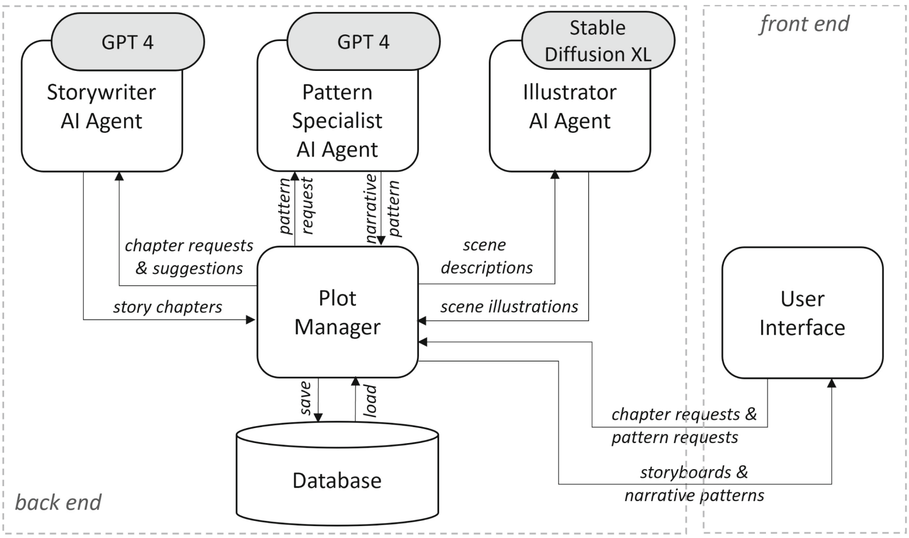

Fig. 1. The architecture of PatternTeller

The AI agents are implemented according to a plugin architectural approach to facilitate the future replacement of the AI agents. As indicated in Fig. 1, in the current version of PatternTeller both the Storywriter AI Agent and the Pattern Specialist AI Agent use the text generation capabilities of the GPT 4 model (accessed via the OpenAI API). The Illustrator AI Agent, on the other hand, is built upon a fine-tuned version of the Stable Diffusion XL model hosted on a private server and accessed through a REST API.

4.2 Pattern Generation

The possibility of generating new narrative patterns from existing stories is a core functionality of PatternTeller. The proposed method explores GPT-4's language modeling capabilities and extensive knowledge of existing stories to derive narrative patterns. Our method is designed to handle two types of input: (1)

plain text summaries of one or more existing narratives; and (2) titles of well-known stories (such as books and films). In both cases, a specialized prompt designed to guide GPT-4 in the process of performing MSG is used (Prompt 1).

Prompt 1. *Generate a generic pattern that describes analogous events related to the narrative stages of the provided narrative texts. Each narrative stage must contain a title starting with"STAGE:", and a short description of the stage starting with "DESCRIPTION:". The description at each stage must consist of a complete action sentence expressing the most specific generalization of the stage events taken from the texts. In addition, character names must be replaced by their roles in the story or character types. Write only the narrative stages.*

Prompt 1 was designed through multiple experiments with GPT 3.5 and GPT 4 to identify a way to apply MSG to find common elements within the input that can be abstracted into broader pattern stage descriptions. The specified format for the output in the prompt plays an important role in ensuring the system can easily interpret the generated patterns.

After generating the pattern as plain text, the system parses it to create the formal representation of the pattern as $P_i = \{S_1, S_2, ..., S_n\}$, where S_i represents a stage of the narrative pattern P_i. Every stage is a tuple $S_i = (t_i, d_i)$, where t_i is the title of the stage, and d_i refers to the description of the stage.

4.3 Story Generation

The proposed story generation process uses the guidance provided by the predefined and user-generated narrative patterns to ensure thematic consistency and structural coherence within the resulting stories. The user plays an essential role in initiating and guiding the story generation process. To begin with, the user provides a premise (R_i), which can be an original idea or an allusion to an existing narrative. Additionally, the user selects the desired narrative pattern P_i to guide the structure of the story.

Based on the provided input, a specialized prompt to instruct GPT-4 on how to compose the story is generated. This prompt incorporates R_i, P_i, and instructions for the LLM to generate the story chapter by chapter. Prompt 2 describes the parameterized structure of this prompt, which was designed and refined through multiple experiments performed with GPT 4.

Prompt 2. *Premise:* $<R_i>$
Stages of the narrative pattern: $<P_i = \{S_1, S_2, ..., S_n\}>$
I'd like you to write a new creative story based on the indicated premise and narrative pattern. For each stage, focus solely on the specific events of that stage and break down the narrative of the stage into a sequence of events, where each event starts with "EVENT X:" and is followed by the narrative of the event. After the description of the event, add a second line starting with "IMAGE X:" containing a short description of an image that illustrates the narrative event. Write only the description of the events and images (do not add comments).

Let's start with the first stage: "$<t_1 \in S_1>$*". After you complete the events for this stage, please stop, and I'll ask you for the next stage, ensuring each stage is described independently. Also generate a new creative title for the story and add it at the beginning of the response starting with "TITLE:".*

A key feature of Prompt 2 is its directive to start by only generating events of the first stage of the narrative pattern. This allows users to actively engage in a step-by-step narrative generation process. At every step, two operations are available for users: (1) Continue Story, which allows users to request the Storywriter AI Agent to write the next chapter for the story (optionally suggesting what should happen in the next chapter); and (2) Regenerate Chapter, which allows users to request the Storywriter AI Agent to write an alternative version of the current chapter. This interactive story generation process allows users to evaluate the content of the story, propose new events, and refine existing ones to compose a story that fits the user's expectations.

The Continue Story operation allows users to actively engage in the narrative composition process by providing suggestions that can influence the direction of the narrative. These suggestions are optional, thus allowing users to either actively shape the story or passively observe the AI Agent's autonomous generation of the story. Internally, a continue prompt (Prompt 3) is used to instruct the AI Agent to generate the next chapter for the story. Whenever a suggestion is provided (U_j), an alternative continue prompt is used (Prompt 4).

Prompt 3. *Now, I'd like you to move to the next stage: "*$<t_i \in S_i>$*". After you complete the events for this stage, please stop, and I'll ask you for the next stage, ensuring each stage is described independently.*

Prompt 4. *Now, I'd like you to move to the next stage: "*$<t_i \in S_i>$*". Please consider this suggestion while writing this stage:* $<U_j>$*. After you complete the events for this stage, please stop, and I'll ask you for the next stage, ensuring each stage is described independently.*

In order to generate stories, PatternTeller explores GPT 4's ability to engage in conversational interactions, which allows a collaborative narrative generation process. The basic structure of this conversation is:

1. PatternTeller: *Prompt* 2
2. GPT 4: *TITLE: story title. EVENT 1: description of event 1. IMAGE 1: description of scene 1. ... EVENT N: description of event N. IMAGE N: description of scene N.*
3. PatternTeller: *Prompt* 3 *or Prompt* 4
4. GPT 4: *EVENT 1: description of event 1. IMAGE 1: description of scene 1. ... EVENT N: description of event N. IMAGE N: description of scene N.*

Messages 3 and 4 of the conversation structure are repeated until the story reaches its end (i.e. when the last stage of the narrative pattern is used to generate the last chapter of the story).

In addition to the Continue Operation, users can refine the narrative by requesting the regeneration of a chapter. The Regenerate Chapter operation removes the current chapter from the story and prompts the AI Agent to generate a new chapter. By exploring the stochastic nature of GPT 4, this operation introduces different variations of story events, allowing users to explore various narrative possibilities until they find one that aligns with their preferences.

4.4 Image Generation

The proposed image generation process explores the recent advancements in text-to-image machine learning models to create visual representations of the story events. In our implementation, we employ the Juggernaut XL model,[1] which is a fine-tuned version of the Stable Diffusion XL model.

The scene descriptions created during the narrative generation process play a crucial role in the image generation process. While directly feeding these descriptions into the Stable Diffusion model would suffice for producing illustrative representations of the narrative events, we implemented some additional prompt optimizations to improve image quality. First, we incorporated an additional user parameter, the illustration style, which allows users to customize the visual aesthetic of the generated images. This parameter is combined with the description of the scenes to form the final prompt provided to the Illustrator AI Agent. The default illustration style of PatternTeller is "Hyperdetailed Photography, Cinematic", which generates aesthetically pleasing and cinematic-looking images. However, users are allowed to modify this style by providing any desired descriptions, such as the style of famous artists (e.g., "painted by Pablo Picasso").

Another prompt optimization technique that we implemented involves the use of a general negative prompt, which aims to guide the model toward generating more accurate images by explicitly identifying undesirable visual elements. The negative prompt was crafted based on weaknesses observed in the model's outputs for narrative events. The negative prompt used by PatternTeller is: "((poorly drawn hands)), ((poorly drawn face)), blurry, ((bad anatomy)), (((bad proportions))), ((extra limbs)), extra fingers, mutated hands, out of frame, (malformed limbs), ((missing arms)), ((missing legs)), ((extra arms)), (((extra legs))), (((long neck)))", where parentheses are used to convey emphasis, with more parentheses representing a greater emphasis on the undesired elements.

The use of parenthesis to increase the emphasis of certain elements is also applied to the scene descriptions. By emphasizing nouns and adjectives, we were able to produce more concise illustrations. The process of automatically adding parenthesis to scene descriptions is implemented using the natural language processing library spaCy, which is able to create linguistic annotations for the identification of noun phrases and adjectives. For each noun chunk identified in the scene description, the algorithm distinguishes nouns and proper nouns (NOUN, PROPN) from their associated adjectives (ADJ) and coordinating conjunctions

[1] https://civitai.com/models/133005/juggernaut-xl.

(CCONJ). Then, the method constructs noun-adjective pairs, mapping each noun into its preceding adjectives, if any. Nouns without adjectives receive a double set of parentheses, signaling a lesser degree of emphasis, whereas noun-adjective pairs are enclosed in triple parentheses for a higher degree of emphasis.

For example, by giving the following scene description: "Eira is shown in her humble cottage, with a small flame dancing on her fingertips, her eyes reflecting curiosity and wonder", our method will produce the following description: "((Eira)) is shown in her (((humble cottage))), with a (((small flame))) dancing on her ((fingertips)), her ((eyes)) reflecting ((curiosity)) and ((wonder))".

Figure 2 shows the images generated for the aforementioned scene description by applying the previously described prompt optimization techniques: V1 is the image based only on the original scene description; V2 combines the original scene description with the default illustration style; V3 combines the original scene description, the default illustration style, and the general negative prompt; and V4 combines all prompt optimization techniques used in V3 with the scene description that uses parentheses to increase the emphasis of certain elements.

V1 V2 V3 V4

Fig. 2. Example of image variations generated by applying different prompt optimization techniques

4.5 User Interface

As illustrated in Fig. 3, the user interface of the system is designed to facilitate both story composition and the creation of new narrative patterns. To compose a new story, users must provide a short premise, select the desired narrative pattern, provide an optional illustration style, and then press the "Generate Story" button on the initial screen of PatternTeller (Fig. 3 (I)). Then, the system transitions to the story composition screen (Fig. 3 (II)), where users can view the generated story's title, the events of the first chapter in a storyboard format, and continue with the story composition process. When creating a new narrative pattern, the user can decide whether to extract the pattern from existing stories or manually create the pattern. To extract it from existing stories, one or more stories must be provided as input. Then, the "Extract Narrative Pattern" button can be used to request the system to generate a narrative pattern (Fig. 3 (III)).

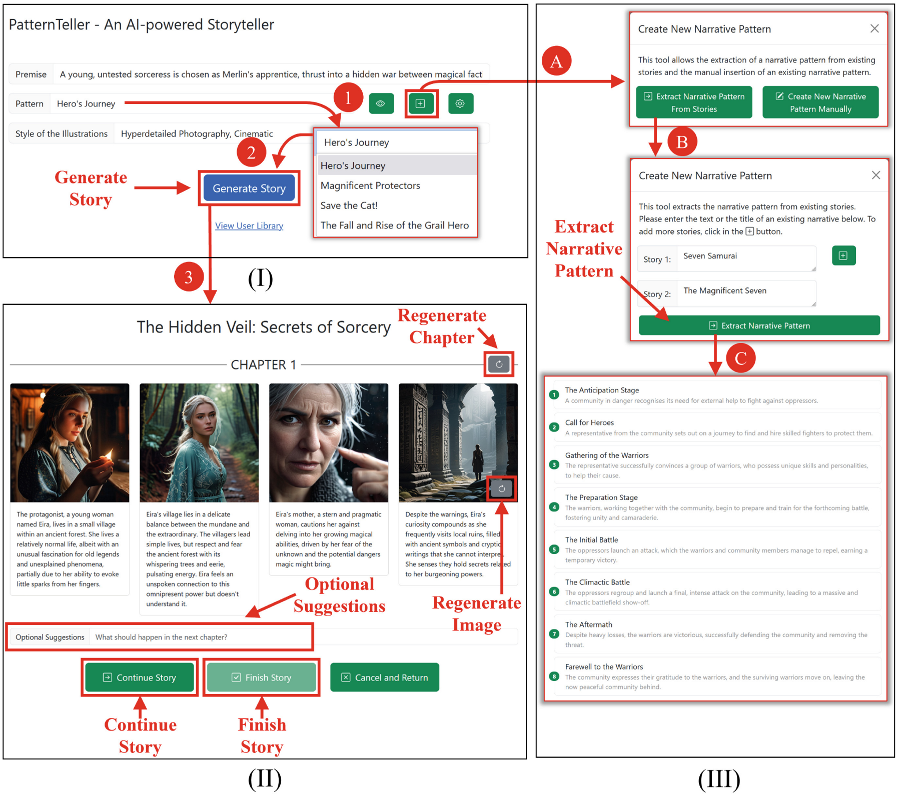

Fig. 3. The user interface of PatternTeller

In the story composition example illustrated in Fig. 3, the user first entered the premise "A young, untested sorceress is chosen as Merlin's apprentice, thrust into a hidden war between magical factions. She must master her chaotic powers under Merlin's cryptic guidance while battling a malevolent force seeking to exploit her potential" (Action 1 in Fig. 3 (I)), and selected *Hero's Journey* as the narrative pattern (Action 2 in Fig. 3 (I)). Then, clicking on the "Generate Story" button (Action 3 in Fig. 3 (I)), prompted PatternTeller to generate a new story titled *The Hidden Veil: Secrets of Sorcery*. After viewing the first chapter, the user has various options: either regenerate the entire chapter by pressing the "Regenerate Chapter" button, or regenerate only the images of certain scenes by clicking on the reload button that becomes visible when the user hovers the mouse over any scene illustration, or continue the story by pressing the button "Continue Story" to generate the next chapter for the story, deciding whether or not to provide suggestions in the "Optional Suggestions" field. The system will consider these suggestions during the next chapter generation or regeneration. The full story generated for this example is available at: https://narrativelab.org/patternteller/view.php?id=293.

In the pattern generation example, also presented in Fig. 3, the user started the process of creating a new narrative pattern (Action A in Fig. 3 (III)) and decided to extract it from existing stories (Action B in Fig. 3 (III)). The titles of two movies were provided by the user as input: *Seven Samurai* and *The Magnificient Seven*. Then, clicking on the "Extract Narrative Pattern" button (Action C in Fig. 3 (III)), triggered PatternTeller to produce the narrative pattern called *Magnificent Protectors*, mentioned in Sect. 3.2. The full description of this pattern is available at https://narrativelab.org/patternteller/pattern.php?id=8.

5 Concluding Remarks

We proposed an AI-powered approach that fulfilled our initial intention of using narrative patterns to promote thematic consistency. However, we soon realized that patterns, besides favoring good authorial practices, served the complementary purposes of *repetition* and *innovation*. As to repetition, the ability to compose following a pattern allowed to retell the same course of events in a different context, as well as to pursue continued production ventures such as franchises and series. As to innovation, creating new patterns permitted to introduce new types, congenial to modern trends, by generalizing two or more popular works, or by entering a single narrative, which may even consist of a tentative text by a beginner author. Once created, a new pattern is ready for story composition to be triggered by a correlated user's premise. These conclusions are in line with the results obtained in our recent work on the use of semiotic relations to compose new narratives by reusing elements from existing stories [25].

Resorting to AI agents brings invaluable benefits for narrative generation when compared to plan-based approaches, since difficult and arduous preparatory formal domain specifications are no longer needed in view of the huge information bases memorized as part of LLMs' training. Users, in turn, are not required to learn any formalism, given that their communication with the prototype interface is exclusively in natural language, exactly as the resulting narratives and patterns are all in idiomatic English - whereas the AI agent is largely tolerant with users' typos and imprecise statements.

And yet much work remains to be done as this project continues. Until now the prototype mainly served to us as proof of concept, for which it has been very useful by giving us a better understanding of how patterns function as defining accessories to types, or to entire genres as happens with the Monomyth. Extensive user studies to evaluate usability and story quality are a short-term necessity that we plan to explore as the next step of this project. Furthermore, a particularly crucial research goal is to revise in depth the way LLMs process the MSG method applied to pattern creation, with the objective of grading the level of analysis so as to improve the capture of commonalities.

As to the occasionally still faulty performance of Stable Diffusion, the AI agent responsible for scene images, future efforts may target further improving the image generation process. A promising solution involves the development of a library with detailed descriptions of characters and their physical attributes. By allowing users to select the protagonists of their stories from this set of predefined characters, the system would be able to provide more detailed descriptions of the scenes to the Stable Diffusion model, which can possibly enhance character consistency across images in the same story.

Lastly, future work may go beyond the linear patterns currently handled by the prototype, in order to treat, for example, network patterns with forks, joins, loops, and distinct mandatory or optional events, thereby increasing the prototype's functionality and enhancing interactivity, by offering to users a succession of multiple-choice decision points.

Acknowledgements. 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.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alabdulkarim, A., Li, W., Martin, L.J., Riedl, M.O.: Goal-directed story generation: augmenting generative language models with reinforcement learning. arXiv preprints arXiv: 2112.08593 (2021)
2. Alvarez, A., Font, J.: TropeTwist: trope-based narrative structure generation. In: Proceedings of the 17th International Conference on the Foundations of Digital Games. FDG '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3555858.3563271
3. Bal, M.: Narratology: Introduction to the theory of narrative. University of Toronto Press (2007)
4. Campbell, J.: The Hero with a Thousand Faces. New World Library (2008)
5. Castricato, L., Frazier, S., Balloch, J., Riedl, M.: Tell me a story like I'm five: story generation via question answering. In: Proceedings of the 3rd Workshop on Narrative Understanding (2021). https://par.nsf.gov/biblio/10249509
6. Cavazza, M., Pizzi, D.: Narratology for interactive storytelling: a critical introduction. In: Göbel, S., Malkewitz, R., Iurgel, I. (eds.) TIDSE 2006. LNCS, vol. 4326, pp. 72–83. Springer, Heidelberg (2006). https://doi.org/10.1007/11944577_7
7. Chu, H., Liu, S.: Can AI tell good stories? Narrative transportation and persuasion with ChatGPT. In: PsyArXiv. PsyArXiv (2023).https://doi.org/10.31234/osf.io/c3549
8. Ciarlini, A.E.M., Casanova, M.A., Furtado, A.L., Veloso, P.A.S.: Modeling interactive storytelling genres as application domains. J. Intell. Inf. Syst. **35**(3), 347–381 (2010). https://doi.org/10.1007/s10844-009-0108-5
9. Ciarlini, A.E.M., Pozzer, C.T., Furtado, A.L., Feijó, B.: A logic-based tool for interactive generation and dramatization of stories. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment

Technology, pp. 133–140. ACE '05, Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1178477.1178495
10. de Lima, E.S., Feijó, B., Barbosa, S.D., Furtado, A.L., Ciarlini, A.E., Pozzer, C.T.: Draw your own story: paper and pencil interactive storytelling. Entertainment Comput. **5**(1), 33–41 (2014). https://doi.org/10.1016/j.entcom.2013.06.004
11. de Lima, E.S., Feijó, B., Casanova, M.A., Furtado, A.L.: Storytelling variants based on semiotic relations. Entertainment Comput. **17**, 31–44 (2016). https://doi.org/10.1016/j.entcom.2016.08.003
12. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 889–898. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1082
13. Feijó, B., de Lima, E.S., Furtado, A.L.: A transdisciplinary approach to computational narratology. In: Proceedings of the XX Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 2021), pp. 1–10 (2021)
14. Frye, N.: Anatomy of Criticism. Princeton University Press (2020)
15. Furtado, A.L.: Analogy by generalization-and the quest of the grail. SIGPLAN Not. **27**(1), 105–113 (1992). https://doi.org/10.1145/130722.130741
16. García-Sánchez, P., Velez-Estevez, A., Julián Merelo, J., Cobo, M.J.: The Simpsons did it: exploring the film trope space and its large scale structure. PLOS ONE **16**(3), 1–28 (2021). https://doi.org/10.1371/journal.pone.0248881
17. Jacobi, J.: The Psychology of CG Jung. Yale University Press (1973)
18. de Lima, E.S., Casanova, M.A., Feijó, B., Furtado, A.L.: Semiotic structuring in movie narrative generation. In: Ciancarini, P., Di Iorio, A., Hlavacs, H., Poggi, F. (eds.) Entertainment Computing – ICEC 2023, pp. 161–175. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8248-6_13
19. de Lima, E.S., Feijó, B., Casanova, M.A., Furtado, A.L.: ChatGeppetto - an AI-powered storyteller. In: Proceedings of the 22nd Brazilian Symposium on Games and Digital Entertainment, pp. 28–37. ACM (2024).https://doi.org/10.1145/3631085.3631302
20. de Lima, E.S., Feijó, B., Furtado, A.L.: Hierarchical generation of dynamic and nondeterministic quests in games. In: Proceedings of the 11th Conference on Advances in Computer Entertainment Technology. ACE '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2663806.2663833
21. de Lima, E.S., Feijó, B., Furtado, A.L.: Video-based interactive storytelling using real-time video compositing techniques. Multimedia Tools Appl. **77**(2), 2333–2357 (2018). https://doi.org/10.1007/s11042-017-4423-5
22. de Lima, E.S., Feijó, B., Furtado, A.L.: Computational narrative blending based on planning. In: Baalsrud Hauge, J., C. S. Cardoso, J., Roque, L., Gonzalez-Calero, P.A. (eds.) ICEC 2021. LNCS, vol. 13056, pp. 289–303. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89394-1_22
23. Soares de Lima, E., Feijó, B., Furtado, A.L., Diniz Junqueira Barbosa, S., Pozzer, C.T., Ciarlini, A.E.M.: Non-branching interactive comics. In: Reidsma, D., Katayose, H., Nijholt, A. (eds.) ACE 2013. LNCS, vol. 8253, pp. 230–245. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03161-3_16
24. de Lima, E.S., Feijó, B., Furtado, A.L.: Procedural generation of branching quests for games. Entertainment Comput. **43**, 100491 (2022). https://doi.org/10.1016/j.entcom.2022.100491

25. de Lima, E.S., Neggers, M.M.E., Feijó, B., Casanova, M.A., Furtado, A.L.: An AI-powered approach to the semiotic reconstruction of narratives. Entertainment Comput. **52**, 100810 (2024). https://doi.org/10.1016/j.entcom.2024.100810
26. de Lima, E.S., Neggers, M.M.E., Furtado, A.L.: Multigenre AI-powered story composition. arXiv preprints arXiv: 2405.06685 (2024)
27. de Lima, E., Furtado, A., Feijó, B., Casanova, M.: Towards reactive failure-recovery gameplaying: the fall and rise of the grail hero. In: Proceedings of the XV Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 2016), pp. 262–271. São Paulo, Brazil (2016)
28. Propp, V.: Morphology of the Folktale. University of Texas Press (1968)
29. Rank, O., Raglan, L., Dundes, A.: In quest of the hero. In: In Quest of the Hero. Princeton University Press (1990)
30. Richmond, C., Schoentrup, D.: TV tropes. https://tvtropes.org/pmwiki/pmwiki.php/Main/Tropes
31. Ryan, M.-L.: Interactive narrative, plot types, and interpersonal relations. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 6–13. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89454-4_2
32. da Silva, F.A.G., Furtado, A.L., Ciarlini, A.E.M., Pozzer, C.T., Feijó, B., de Lima, E.S.: Information-gathering events in story plots. In: Herrlich, M., Malaka, R., Masuch, M. (eds.) ICEC 2012. LNCS, vol. 7522, pp. 30–44. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33542-6_3
33. Snyder, B.: Save the Cat! The Last Book on Screenwriting You'll Ever Need. Michael Wiese Productions (2005)
34. Uther, H.: The Types of International Folktales: A Classification and Bibliography: Based on the System of Antti Aarne and Stith Thompson. Finnish Academy of Science and Letters (2011)
35. Vogler, C.: The Writer's Journey. Michael Wiese Productions (2007)
36. Värtinen, S., Hämäläinen, P., Guckelsberger, C.: Generating role-playing game quests with GPT language models. IEEE Trans. Games **16**(1), 127–139 (2024). https://doi.org/10.1109/TG.2022.3228480
37. Wang, S., Durrett, G., Erk, K.: Narrative interpolation for generating and understanding stories. arXiv preprints arXiv:2008.07466 (2020)
38. Xiang, J., et al.: Interleaving a symbolic story generator with a neural network-based large language model. In: Proceedings of Tenth Annual Conference on Advances in Cognitive Systems (2022)
39. Xie, K., Riedl, M.: Creating suspenseful stories: Iterative planning with large language models. In: Graham, Y., Purver, M. (eds.) Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2391–2407. Association for Computational Linguistics, St. Julian's, Malta (2024). https://aclanthology.org/2024.eacl-long.147
40. Xu, P., Patwary, M., Shoeybi, M., Puri, R., Fung, P., Anandkumar, A., Catanzaro, B.: MEGATRON-CNTRL: controllable story generation with external knowledge using large-scale language models. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2831–2845. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.226
41. Yang, D., Zhou, Y., Zhang, Z., Li, T., LC, R.: AI as an active writer: interaction strategies with generated text in human-AI collaborative fiction writing. In: Smith-Renner, A., Amir, O. (eds.) Joint Proceedings of the IUI 2022 Workshops, pp. 56–65. CEUR Workshop Proceedings (2022). http://ceur-ws.org/Vol-3124/

42. Yang, K., Tian, Y., Peng, N., Klein, D.: Re3: generating longer stories with recursive reprompting and revision. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 4393–4479. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022). https://doi.org/10.18653/v1/2022.emnlp-main.296
43. Yuan, A., Coenen, A., Reif, E., Ippolito, D.: Wordcraft: story writing with large language models. In: Proceedings of the 27th International Conference on Intelligent User Interfaces, IUI '22, pp. 841–852. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3490099.3511105