

Toward a Computational Framework of Suspense and Dramatic Arc

Brian O’Neill and Mark Riedl

School of Interactive Computing
Georgia Institute of Technology
{boneill, riedl}@cc.gatech.edu

Abstract. We propose a computational framework for the recognition of suspense and dramatic arc in stories. Suspense is an affective response to narrative structure that accompanies the reduction in quantity or quality of plans available to a protagonist faced with potential goal failure and/or harm. Our work is motivated by the recognition that computational systems are historically unable to reliably reason about aesthetic or affective qualities of story structures. Our proposed framework, Dramatis, reads a story, identifies potential failures in the plans and goals of the protagonist, and computes a suspense rating at various points in the story. To compute suspense, Dramatis searches for ways in which the protagonist can overcome the failure and produces a rating inversely proportional to the likelihood of the best approach to overcoming the failure. If applied to story generation, Dramatis could allow for the creation of stories with knowledge of suspense and dramatic arc.

Keywords: Affective computing, suspense, dramatic arc, narrative cognition

1 Introduction

Narrative as entertainment, in the form of oral, written, or visual storytelling, plays a central role in many forms of entertainment media, including novels, movies, television, and theatre. One of the reasons for the prevalence of storytelling in human culture may be due to the way in which narrative is a cognitive tool for situated understanding [1, 2]. This narrative intelligence is central in the cognitive processes that we employ across a range of experiences, from entertainment contexts to active learning. Expert storytellers who craft narratives for entertainment – films, novels, games, etc. – often structure their narratives to elicit an emotional response from the viewer, reader, or player. The concept of the dramatic arc, identified by Aristotle [3], is one common pattern of emotional impact on an audience.

The construction of novel quality stories is a challenging task, even for humans. For more than 30 years, computer scientists have been trying to answer the question of whether, and how, intelligent computational systems can create stories from scratch. Computational story generation systems typically take one of two approaches. In a search-based approach, the system searches the space of possible stories to find the best story according to a heuristic. In the adaptation-based approach, systems take one or more existing stories, which are altered or recombined

into new narratives. See Gervás [4] for a summary of the history of story generation. We make the observation that, to date, story generation systems have been unreliable when it comes to creating novel stories with dramatic structure. Simply put, story generation systems do not have sufficient understanding of story aesthetics nor how story structure affects emotional change in an audience.

In this paper, we explore what it would take to make a story generation system be able to understand the concept of dramatic arc. There are many ways to produce dramatic arc in a story [5]. One such approach is to make stories suspenseful; Abbott describes suspense and surprise as the two things that “give narrative its life” [6]. Despite the importance and prevalence of suspense as a storytelling tool, there has been little investigation of computational techniques for generating or understanding suspense. The most relevant work has been in discourse generation: deciding what to tell and what to leave out when telling a pre-existing story [7]. However, the question of how to generate interesting stories from scratch remains open.

This paper describes preliminary work toward *Dramatis*: a framework for the detection of suspense, as part of dramatic arc. The proposed system “reads” elements of a story one at a time and generates a suspense response when appropriate. While we do not directly address the question of how to computationally generate suspenseful stories, our work on suspense detection will be used to heuristically guide search-based story generation processes. In the remainder of this paper, we discuss related work in narratology and computational approaches to suspense and narrative. The next section describes *Dramatis* and walks the reader through an example using a recent film. Finally, we discuss why we believe this framework is plausible and describe future work in the implementation and testing of the framework.

2 Related Work

Suspense occurs in an audience – the reader or watcher of a narrative – when the audience perceives that a protagonist is faced with the possibility of an undesirable outcome. Gerrig and Bernardo suggest that one method used by authors to make readers feel suspense is to reduce the quantity and/or quality of plans available to the protagonist for avoiding an undesirable outcome [8]. They suggest that readers act as problem-solvers on behalf of the protagonist and when readers can only devise low-quality plans, or struggle to come up with any plans for a hero to escape the predicament, the perception of suspense will increase. In these studies, they found that readers reported higher suspense levels when story excerpts suggested potential escapes and then quickly eliminated them, thus reducing the quantity of available plans for the protagonist.

Branigan [5] suggests that suspense can occur when the audience knows more than the protagonist, particularly about the possibility of undesirable outcomes. He also points out that feelings of suspense can be intensified depending on the audience’s affinity for the character in question. Suspense has also been described as a lack of closure within a narrative [6]; authors manipulate readers by appearing to satisfy the need for closure, only to take it away.

Readers continuously make inferences about aspects of the story which have not

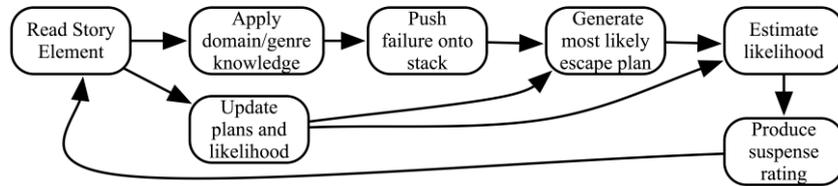


Figure 1. This flowchart represents the process used by Dramatis to move from reading part of a story to producing a suspense rating.

been explicitly stated [9]. These inferences can be divided into two classes. *Online inferences* can be made with little effort while reading, while *offline inferences* require that the audience is given time to reason. Online inferences include the recognition of causal antecedents and superordinate goals. Offline inferences include identifying subordinate goals, such as recognizing the lesser goals of a character or the plan of action used to achieve some state. Both classes of inference are important in recognizing when readers feel suspense.

The Suspenser system [7] computationally attempts to tell pre-existing stories in a suspenseful way. Suspenser tries excluding different sets of events from the discourse to maximize a suspense rating. The level of suspense of any telling of a story is measured by generating all possible plans a protagonist might have and taking the ratio of failed plans to successful plans. We are not seeking to modify existing stories, but rather detect suspenseful points in the discourse, as given. Additionally, in order to preserve cognitive plausibility our metric for suspense is not based on an exhaustive search of the plan-space.

3 Dramatis Framework

We present a framework for Dramatis: a suspense detection system based on the correlation between perceived likelihood of a protagonist’s failure and the amount of suspense reported by the audience. Dramatis reads in elements of the story one at a time. The system attempts to predict failures in the protagonist’s goals and plans, making predictions only when the pacing of the story affords time to engage in offline processing. Predictions are made using knowledge of the story domain and genre, and knowledge of the goals and plans of the characters in the story when available.

When a failure prediction is made, Dramatis must predict how likely it is that the failure will actually occur. Instead of exhaustively sampling the space of possible plans for those that fail and those that succeed, we invert the process by searching for the most likely plan that will avoid failure, which we call the *escape plan*. The system searches for the most likely escape plan and estimates the likelihood of the plan’s success under the assumption that one only needs to find the best way to escape the failure given existing knowledge about the world. Dramatis uses the likelihood of successful escape to determine a suspense rating for the given point in the story;

suspense is inversely proportional to the likelihood of failure escape. The system tracks the generated escape plan, and as the story progresses and new information is added, it recalculates the likelihood or regenerates the plan, if necessary. The flowchart in Figure 1 shows the complete suspense detection process. We highlight the key points of the framework below.

3.1 Reading a Story Element

Dramatis reads in story elements one at a time. We do not address any aspect of natural language understanding or computer vision necessary to literally read or view a creative artifact such as a story or film. Instead, the system will read in an annotated version of a story or film script, which is discretized into “time-slices.” Each time-slice contains details about the characters involved in the current scene that are known to the audience (and may or may not be known to the protagonist), the location of the scene, symbolic representations of both the content of the dialogue, the actions performed by the characters and other non-agent items in the scene, the pacing of the scene (indicating whether the audience would have time to do offline processing), and non-diegetic information such as background music and camera angles used (as both of these are used by directors to manipulate audience emotion).

3.2 Searching for Failure

Dramatis searches for a failure when cued to do so by the most recently read time slice. This cue may come from non-diegetic information, such as ominous music or from new information about the narrative. When the audience is given new information that is not addressing an existing potential failure, Dramatis searches for a new potential failure that could disrupt the protagonist’s pursuit of his goal. We have identified three cases. First, an opposing plan or goal is given as part of the time-slice. For example, a character being told he will be shot provides an opposing goal of that character’s death. Second, the system applies relevant knowledge, such as domain and genre information, to determine a potential failure. This knowledge takes the form of scripts and schemas. For example, the knowledge that a character being poisoned typically leads to that character getting sick and then dying would be contained in a script. We do not explicitly search for this script; rather, it is retrieved from a collection of scripts indexed by character goals and elements present in the scene. Third, when Dramatis knows the protagonist’s plan and the plans of an antagonist character, it may determine that these plans will require actions that are mutually exclusive, such that only one of the characters’ plans can succeed. In this case, the other character’s plan is stored as an opposing plan.

The scripts and plans retrieved during the search for failure are represented as plan networks [10]. Plan networks allow scripts to be represented as a set of possible sequences of discrete events. We have modified plan networks so that they contain explicit information about the causal relationships between the events listed. A causal relationship exists between two events when one event affects change on the world necessary for a later event to occur. Thus, each event in the script is described by a set

of preconditions and effects where preconditions describe the state of the world that must exist before this event can occur and the effects describe how the world will be different after the event occurs. The preconditions and effects are given as a set of symbolic predicates that describe the world state. This approach to describing actions and their preconditions and effects is common in planning problems, but can easily be transferred to scripts as a form of meta-knowledge that enables an agent to reason about scripts and plans.

Dramatis may generate many hypothetical failures; the failure with the most negative utility – the one that would be most undesirable for the protagonist – is selected. Dramatis maintains a failure stack to track the obstacles to the protagonist's goals in the case that multiple failures arise. We make the simplifying assumption that the audience responds emotionally to the most immediate potential failures first.

3.3 Computing Likelihood of Escape

Once Dramatis has identified a new potential failure, it uses its knowledge of the protagonist's goal and the potential failure to search for an *escape plan* – a plan that enables the protagonist to avoid the undesirable outcome, while still allowing the protagonist to proceed toward his original goal. We generate an escape plan for the purpose of identifying the likelihood of the protagonist's escape, which is correlated with the suspense felt by a human audience.

Dramatis uses a form of Heuristic Search Planning (HSP) called set-additive HSP, which supports non-uniform action costs [11], to find an escape plan for the protagonist. HSP is a type of informed state-space search that uses a relaxed form of the planning problem to efficiently estimate the distance to a goal state. The distance to the goal state is the estimated sum of the action costs for reaching a goal state. To compute the escape plan, we use the current state of the story as the initial state, and the goal situation is any state in which any of causal relations in the script or plan identified during failure search becomes negated, so long as that negated state is not part of the final state of the script or plan – that is, the negated state should not assist in completing the script/plan. In this sense, the computation of the escape plan is a form of single-shot adversarial planning. The extent to which recursive adversarial planning is required has yet to be fully determined, however it is known that human processing of suspense is highly bound by time and effort.

Dramatis uses the likelihood of the first found escape plan as an indication of level of suspense. To ensure that the first complete plan found is also the most likely to succeed in averting the potential failure, we use likelihood of each action as action cost in set-additive HSP. Specifically, action cost is computed as a function of (a) the mental effort of retrieving the action from a computational memory, (b) the overall likelihood of that event succeeding given the state it will be executed in, and (c) the time remaining before the opposing plan/script is completed. MacLeod and Campbell [12] note that humans perceive events that can easily be retrieved from memory as being more likely to occur. When applied to the generation of escape plans, this suggests that actions more readily retrieved will be perceived as more likely to succeed. We intend to use a computational model of memory from which events are more easily retrieved when their elements, such as characters and objects involved or

where the event occurs, are closely related to the same elements in recently mentioned events [13]. Using this approach, Dramatis can account for genre effects and for an audience's tendency to forget key details that have not been addressed recently. The overall likelihood of event success is calculated as a function of the state of the world. Humans learn these likelihoods through experience. An analogue in computational agents is a Q-function, a table look up of utility – in our case likelihood – given a state-action pair. In order to compute suspense, Dramatis must be bootstrapped with knowledge that would otherwise be learned over a lifetime of experiences. Finally, time is incorporated into the likelihood calculation based on the remaining steps in the opposing plan or script, based on the assumption that as the time available to avert a failure decreases, the quantity of available escape plans also decreases. Therefore, as the number of steps remaining decreases, the suspense level increases.

The system then applies the likelihood of the success of the escape plan to the computation of a suspense rating. Recall that as the likelihood of escape increases, suspense is expected to decrease. For each potential failure on the failure stack, Dramatis has found an escape plan and determined its likelihood. The suspense rating is proportional to the sum of the likelihoods of each escape plan, factoring in the affinity the audience feels for the protagonist, and the severity (in terms of negative utility) of the failures. Thus, because utility is negative, as likelihood goes down, suspense goes up. The likelihoods of escape plans for failures that are lower on the stack are discounted when calculating this rating. This final aspect exists in order to compute suspense with emphasis on the failure currently being faced, rather than older failures that may not be at the forefront of the story at the moment.

3.4 Updating Suspense

Once Dramatis has found an escape plan and calculated the suspense rating, it continues to the next time slice in the story. As it reads from this point, there are four possibilities for what occurs next: (1) The next step in the projected escape plan is executed; (2) The projected escape plan has been eliminated as an option by newly introduced information; (3) a different escape plan has been explicitly provided by the story and potentially acted upon; or (4) some different action occurs that does not fit with the projected escape plan, and may be part of some other escape plan.

In the first case, the projected escape plan appears to be correct. Dramatis keeps the projected escape plan and updates its estimate of the plan's likelihood of success with the knowledge that this step of the escape plan has been executed. The updated likelihood estimate leads to a new suspense level estimate. If the failure has now been completely avoided, it is removed from memory after the suspense level has been recalculated, and the system reverts to having no failure predicted.

In the second case, something has occurred to eliminate the current escape plan as an option. This might be because some new knowledge was provided, such as one of the escape plan prerequisites not being available. In this case, Dramatis uses its updated knowledge of the world to search for new escape plans using the same potential failure as before.

In the third case, some aspect of the newly read event specified the escape plan that the protagonist is going to use. This plan is stored, replacing the escape plan that was

predicted in some earlier iteration. Dramatis then estimates the likelihood of success of this new plan, and determines the suspense level based on this new information.

In the final case, an action occurred that does not fit with the predicted escape plan, but is not direct conflict with that escape plan either. This action may be the first step of a different escape plan that the protagonist is using instead. Given this new action, Dramatis searches for an escape plan that is consistent with the new action. Dramatis calculates the likelihood of the new escape plan and recalculates the suspense level.

3.5 Knowledge Structure

Given this process for detecting suspense, what knowledge must the system have available to it, and what knowledge must it track throughout? From the outset, Dramatis must be provided with background knowledge. This knowledge includes domain knowledge for the purposes of planning, identifying potential failures, and estimating the likelihood of a plan's success. The domain knowledge comes in the form of scripts and action definitions. The scripts, represented as plan networks [10], are used for retrieving potential failures, while the action definitions inform the heuristic search planning.

Dramatis also tracks knowledge about the story through successive iterations. The system keeps track of character plans and goals, particularly those of the protagonist, whenever they are provided. Additionally, these goals include maintenance goals such as Avoid-Dying. Character plans and goals are used to predict possible failures and to identify conflicting plans. The system also tracks the potential failures that it predicts in the protagonist's plan. Finally, when the system predicts an escape plan from these failures, it is stored for later comparison. The system keeps track of whether its predicted escape plans are being used or being eliminated, as this will affect the audience suspense level.

3.6 Example

Consider the following example taken from the 2006 film *Casino Royale*¹. In the film, the protagonist, James Bond, must play a high-stakes poker game in order to bankrupt international terrorist Le Chiffre and bring him to justice. Prior to the scene in question, the audience is aware of Bond's goal. At the beginning of the example, the following are known: (a) Le Chiffre intends to win this game and will kill Bond to do so if necessary, (b) Bond-ally Vesper is watching the poker game, and (c) Bond has a chip in his arm that allows British intelligence (MI6) to monitor his vital signs. Bond has the goal of winning the game as well as the maintenance goal of staying alive. In the preceding time slice, unbeknownst to Bond, his drink has been poisoned.

For the purposes of the example, the first time slice processed by Dramatis contains the following information: Bond drinks from his glass, Le Chiffre stares expectantly at Bond, and ominous music begins to play. The time slice also indicates time to think, affording time for offline inferences to be made. The music cues

¹ *Casino Royale* is copyrighted by EON Productions, Columbia Pictures, Danjaq, and United Artists.

Dramatis to search for a potential failure and Dramatis retrieves a script about being poisoned. The poisoned script has the effect of harming Bond, which is antithetical to Bond's maintenance goal. Further, dying is the most dire possible failure state in terms of negative utility. With a potential failure found, Dramatis searches for escape plans as part of determining the audience suspense level. Suppose that the first escape plan generated is "Bond recognizes poison; Bond vomits poison." This plan prevents Bond's death and is determined to have a high likelihood of success. Dramatis thus computes a low suspense level based on this plan for overcoming the failure.

In time slice 2, Bond recognizes that he has been poisoned. The potential failure – Bond's death – has not changed and the escape plan found in the previous iteration seems to hold. As the scene continues, Bond leaves the poker game and goes to the bathroom to vomit, and Vesper notices Bond's departure and appears alarmed.

The next time slice in which Dramatis is afforded time to make offline inferences (Time Slice 5) reveals that Bond is staggering to his car, and still being affected by the poison, despite the earlier attempt to overcome the problem. The update loop rejects the previous escape plan and generates a new escape plan: "Go to doctor; Get help from doctor." Dramatis calculates the likelihood of this plan, finding it to be lower than the previous escape plan; the suspense level, therefore, increases.

In time slice 6, Bond pulls a medical kit and telephone out of his car, and activates the chip in his arm. Dramatis enters the update loop, and sees that Bond is acting on a different plan than the one predicted during the previous iteration. It generates a new escape plan: "Identify antidote; Bond takes antidote." Now that Dramatis knows about the medical kit, it finds this escape plan to be very likely. As the likelihood increased, the suspense level decreases.

In the seventh time slice, MI6 agents, monitoring his internal sensor, call to tell Bond that he will die in the next two minutes, and that they cannot identify an antidote. The escape plan remains intact, but this new information significantly decreases the estimate that the escape plan will succeed. Dramatis computes an increase in suspense. In time slice 8, Bond goes into cardiac arrest, resulting in the generation of a new escape plan: "Bond applies defibrillator; Bond restarts heart." With the new plan and the new information about the imminence of Bond's death, Dramatis finds this plan less likely, and the suspense level increases again.

Bond pushes the button on the defibrillator in the ninth time slice, but there is no effect. He keeps pushing the button, but to no avail. The pacing is such that there is no time for offline reasoning; Dramatis does not update until the next time slice, when it is revealed that a cord has become detached from the defibrillator. The previous escape plan is updated: "Bond re-attaches cord; Bond restarts heart."

In the subsequent time slice, Bond passes out before he can reattach the cord. Dramatis again sees that its most recent plan has been eliminated. It generates a new plan, which it calculates to be even less likely: "Someone sees Bond; They re-attach the cord; They restart his heart." As this is very unlikely (Dramatis has forgotten about Vesper due to the time that has passed since Dramatis was last aware of the character), the suspense becomes very high at this point.

Vesper appears at the car in the last time slice, as the audience hears Bond's heart start to fail. Dramatis recalculates the previous plan's likelihood with the knowledge that someone, Vesper, has appeared. With someone else actually present, the plan's likelihood of success increases, so the suspense level decreases. Finally, Bond wakes

up, and the failure has been averted. From this point, Dramatis would search for possible failures in his plan to defeat and capture Le Chiffre.

4 Discussion and Future Work

We have proposed a framework, Dramatis, for the detection of suspense as part of dramatic arc in a story. The Dramatis approach to suspense is based on Gerrig and Bernardo's suggested approach to suspense – the reduction in quantity or quality of escape plans available to the protagonist, from the point of view of the reader [8]. In addition, we base our approach in the literature of narrative cognition and narratology. With this knowledge, we believe that Dramatis is a reasonable and cognitively plausible approach to the detection of suspense. The approach used by Dramatis to find the most likely escape plan to succeed is based on psychological studies of human perception of likelihood [12], as is our association of perceived likelihood with the level of suspense felt by a human audience [8]. Further, this search for plans is limited in depth and time in order to simulate the cognitive limits faced by human readers. Finally, Dramatis only conducts this search when a human audience would have sufficient time to reason.

We note some other potential benefits of our approach. First, the Dramatis framework accounts for suspense due to forgetting. It is often the case in narratives that clues about how a protagonist will escape are provided early – a gadget Bond receives from MI6, Vesper taking notice of Bond's condition, etc. – with the intention that the audience momentarily forgets. By linking escape plan generation to a computational model of memory based on concept activation [13] and priming [12], we believe we can reproduce event ordering effects on suspense ratings. Second, we believe we can account for genre effects; actions that are highly related to the genre will be more strongly activated than actions that are less associated with the genre.

Dramatis produces a set of suspense ratings for the points in a single story where the audience is given time to reason, or make offline inferences. The suspense ratings produced by Dramatis are relative to other points in the story and cannot be used to make comparisons across several stories. While the curves produced by Dramatis for two different stories can reasonably be compared, the exact numbers produced are not comparable across stories. Likewise, relative change in Dramatis suspense ratings can be compared to relative change in human suspense ratings. However, it is difficult to draw conclusions from comparing exact numbers between computational systems and humans, unless Dramatis has knowledge equivalent to human knowledge.

In the future, we plan to implement the Dramatis framework and apply the suspense detection process to annotated time-slice replications of actual stories and films other than the above example. Additionally, we intend to compare the suspense ratings produced by Dramatis to those given by a human audience as a means of validating the system output. We assert that it is not necessary that the ratings provided by Dramatis exactly match those of the human audience as long as the evaluation shows that the relative change in suspense levels are comparable.

As a practical application, we plan to apply Dramatis to the field of story generation. Story generation is the problem of finding a sequence of events that meets

a given set of storytelling principles and aesthetic criteria, and can be told as a story. Search based story generation techniques such as [14], in particular, solve the story generation problem by adding and ordering events until the criteria are met. However, existing story generation systems have not been reliable at producing stories with dramatic structure, which can come from suspense, because of a lack of computational models of story aesthetics. Dramatis can be viewed as a domain-specific heuristic for story structure quality based on a computational model of perception of suspense. Thus, when applied to search-based story generation systems, we hypothesize that we can increase the overall subjective rating of computer-generated stories. To invoke Dramatis as a heuristic, one could provide an ideal target suspense curve. Dramatis would then “read” stories generated by a search-based story generation system and compute the mean error between Dramatis’ resulting suspense curve and the ideal curve. While this “guess-and-check” method is not ideal, it provides a first step toward informing story generation systems about aesthetics and the cognitive principles of affect. The application of a model of dramatic arc and narrative aesthetics to story generation systems will lead to the creation of stories that are capable of eliciting emotional responses in human readers.

References

1. Bruner, J.: The Narrative Construction of Reality. *Critical Inquiry*. 18, 1--21 (1991).
2. Gerrig, R.J.: *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. Yale University Press, New Haven (1993).
3. Aristotle: *The Poetics* (T. Buckley trans.). Prometheus Books, Buffalo, NY (1992). (Original work published 350 B.C.E.).
4. Gervás, P.: Computational Approaches to Storytelling and Creativity. *AI Magazine*, 30(3), 49--62 (2009).
5. Branigan, E.: *Narrative Comprehension and Film*. Routledge, New York (1992).
6. Abbott, H.P.: *The Cambridge Introduction to Narrative*. Cambridge University Press, Cambridge (2008).
7. Cheong, Y.-G.: *A Computational Model of Narrative Generation for Suspense*. Doctoral dissertation, North Carolina State University (2007).
8. Gerrig, R.J., Bernardo, A.B.I.: Readers as Problem-Solvers in the Experience of Suspense. *Poetics*. 22, 459--472 (1994).
9. Graesser, A.C., Singer, M., Trabasso, T.: Constructing Inferences During Narrative Text Comprehension. *Psychological Review*. 101, 371--395 (1994).
10. Orkin, J.D.: *Learning Plan Networks in Conversational Video Games*. Masters of Science thesis, Massachusetts Institute of Technology (2007).
11. Keyder, E., Geffner, H.L.: Heuristics for Planning with Action Costs. In: Borrajo, D., Castillo, L., Corchado, J. (eds.) *Current Topics in Artificial Intelligence*. LNCS, vol. 4788, pp. 140--149. Springer, Heidelberg (2007).
12. MacLeod, C., Campbell, L.: Memory Accessibility and Probability Judgments: An Experimental Evaluation of the Availability Heuristic. *Journal of Personality and Social Psychology*. 63, 890--902 (1992).
13. Niehaus, J.: *Cognitive Models of Discourse Comprehension for Narrative Generation*. Doctoral dissertation, North Carolina State University (2009).
14. Riedl, M.O., Young, R.M.: Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research*. 39, 217--267 (2010).