Introducing Story Sampling: Preliminary Results of a New Interactive Narrative Evaluation Technique

Ben Samuel, Josh McCoy, Mike Treanor, Aaron A. Reed, Michael Mateas, Noah Wardrip-Fruin
University of California Santa Cruz, Expressive Intelligence Studio
American University, Washington DC
{bsamuel, mccoyjo, aareed, michaelm, nwf}@soe.ucsc.edu
treanor@american.edu

ABSTRACT

The non-linear nature of interactive narratives makes it difficult for authors of these systems to anticipate how players will interact with them. This paper presents the technique of *story sampling* for assisting in the authoring and evaluation of interactive narratives. An example of a potential application of story sampling is presented using data from the interactive narrative *Prom Week*, and authoring insights gained from this analysis are shared.

Categories and Subject Descriptors

K.8.0 [Personal Computing]: General – Games. I.2.4 [Artificial Intelligence]: Knowledge Representation Formalism and Methods – Representations (procedural and rule-based).

General Terms

Design

Keywords

Game design, social simulation, interactive narrative, story sampling, datamining, play traces, game user experience.

1. INTRODUCTION

Interactive narratives are complex systems, and it can be very difficult for creators of these experiences to anticipate how users will ultimately interact with them. Although the nature of the interaction changes from game to game, player interaction in interactive narratives leads to branching, non-linear, or otherwise dynamic ordering of narrative content. Content creators authoring for these types of experiences likely have narrative goals they are trying to achieve, but the dynamism of the content makes authoring for these goals a challenge. Without the privilege of seeing the story ground out on the narrative level for each player, it is difficult to ascertain the shape of the narrative experience being created. Data mining and visualization techniques are commonly used in games to understand how players are interacting with a system, and many valuable insights can be gleaned from their use. However, these techniques only go so far in addressing the above issues in the domain of interactive narrative because they lack narrative specificity: they don't provide a feel for what is going on in the story.

This paper presents the technique of *story sampling* for assisting in the authoring and evaluating of interactive narratives. Story sampling, when used in conjunction with traditional data mining and visualization techniques, can help provide some of that missing narrative specificity. The technique can be broken down into four steps.

Step One: Generate play traces. These might be created by testers during a game's beta period, or be logged by players post release. Regardless of where they come from, it is important that they provide enough details to reconstruct the player experience on the narrative, either directly or through post-processing.

Step Two: Filter the play traces from step one, identifying traces that represent (potentially) interesting or problematic patterns of player experience. For example, if concerned about the amount of repetition of dialog in a system, find traces that contain repetition in dialog. There are myriad proven data mining and machine learning techniques for filtering data, such as clustering algorithms, which will not be discussed in this paper.

Step Three: Take the play traces from step two and organize them into *story beats*. By story beat, we refer to "the smallest unit of character performance that changes the story" [10]. Whereas the play traces from step one are likely machine readable, it is important that these story beats, be they lines of character dialogue, narration, animations, short videos, or otherwise, are human readable. This likely involves the construction of an additional system to automate the beat identification process. Doing this extra work to represent traces as sequences of human readable story beats is vital for allowing authors to make value judgments about the quality of story sequences.

Step Four: Have humans (e.g., authors, game designers, players) look at the story beats generated from step three and make judgments about the quality of the system as a whole. Use those judgments to inform future authoring and system design.

The authors have found that story sampling can be a very useful approach for identifying and fixing issues with an interactive narrative system, as it helps to identify the causes of a system's problems, not just their existence. Moreover, story sampling can reveal "false positives" as well, e.g. situations that might appear undesirable after running through the filtering of step two, but in actuality are fine, or even desirable, when presented as story beats.

We discuss an example of how story sampling might be used to evaluate an interactive narrative. The data for this evaluation comes from *Prom Week*, a game created by the authors that has been generating user playtrace data since February 2012. Though previous publications can provide the reader with a description of *Prom Week* and *Comme il Faut* (*CiF*) [8, 9], the social AI system

that drives the game, it is worthwhile to briefly describe the authoring process for *Prom Week*, which implements a novel approach for interactive story generation and lends itself well to story sampling. Authoring done in *Prom Week* is retargetable; dialogue is not written with specific characters in mind, but rather for specific social states. Any pair of characters can theoretically engage in any authored conversation, as long as the social circumstances that are the preconditions for that conversation hold true for those characters. These conversations, called *instantiations* since they can be instantiated in a wide array of contexts, are *Prom Week*'s story beats. This approach helped address the authorial burden commonly present in interactive narrative experiences, but made it difficult to evaluate stories generated during player interaction. Although traditional data mining provided some good insights, we found it didn't fully support us in understanding the space of dynamic stories generated through player interaction. Story sampling enabled us to get a better view on actual player experience, and consequently, better identify strengths and weaknesses of the system.

2. RELATED WORK

Data mining and visualization have been used in a variety of game design contexts. Heat maps of the first-person shooter *Halo 2* were generated via user tests while the game was under development [5, 15] and identified the primary areas where players were dying, leading to discoveries of unintentional difficulty spikes. This is similar to our work in that data mining led to the discovery of problem areas. However, where a heatmap was a very effective visualization tool for a game that heavily relies on spatial navigation, it is less applicable to the domain of interactive narrative, where physical movement tends to be less important than story choices made.

Visualizations of the paths through specific branching narratives exist as well [6]. In some narrative authoring environments, the visualization is heavily integrated into the authoring process [1, 2]. Though these techniques can answer questions regarding how many paths a player can take in an interactive narrative, they still do not address questions pertaining to the narrative quality of any individual branch.

Large amounts of high-dimensional data is generated by players playing video games. To learn something meaningful or to predict based on this data, many research projects use machine learning and datamining techniques [4, 11, 12, 16] . Algorithms that reduce the dimensionality of player-derived data have been used to find play patterns in *Tera*, *Battlefield 2* and *Tomb Raider* [3, 13]; and determine types of players in serious games [7]. In this work, we buttress the weaknesses of these purely computational approaches by strategically integrating human authors via story sampling.

3. STORY SAMPLING EXAMPLE: REPETITION OF DIALOGUE

Repetition of text is often considered undesirable in games, particularly when the repetition comes in the form of character dialogue. Human players are very good at identifying when text has been repeated, more so than recognizing repetition in animation; this is one reason why games often re-use the same animations throughout, but will leverage herculean authoring efforts to try to produce novel dialogue content. In the domain of animation, the technique of retargeting gets more use out of any single animation by having the same animation be usable by multiple models [14]. Although there are perhaps as many ways to gracefully handle repetition of dialogue as there are interactive narrative systems, it is common across many systems for authors and game-designers to be unsure of the narrative effects dialogue repetition is having on their game. Story sampling is a tool that can provide a view into this often opaque quality.

As detailed above, *Prom Week*'s dialogue system was inspired by the notion of retargeting. For example, an instantiation in which two characters break up with each other could be retargeted to be played out with any two characters that are currently dating. This technique of retargeting dialogue reduces authorial overhead by quite a bit; authors need only write a single instantiation instead of writing a different break up scene for every possible combination of characters, which in *Prom Week*'s case of having 18 characters would be 306 different variations of just this single scene. The tradeoff is that there is potential for large amounts of dialogue repetition.

We attempted to mitigate the problems of repetition by making the dialogue templated. Some elements or "locutions" of these templates are purely pragmatic, such as having characters refer to each other by their correct names and gender pronouns. Some were meant to introduce some variation in the scene, such as the "random" locution in which authors specify a variety of text snippets of which one is chosen at random. Others allow for the personalities of the characters involved to shine, such as character-specific locutions (every character has unique greetings, affirmations, and insults, for example). In *Prom Week*, every instantiation, and the effects it has on the social state (e.g., after a break up scene the two characters involved are no longer dating) is placed into *CiF*'s Social Facts Database (SFDB). Characters can reference the events in the database in later exchanges; for example, two characters can reminisce about a positive interaction they shared earlier in the game. Since both the identity of the characters speaking this dialogue, and the contents of the SFDB are dynamically determined through gameplay and thus unknown at authoring time, it is difficult to know if the actual stories generated by play are in line with the authorial intent behind the instantiation's creation.

We knew some dialogue repetition would be present in *Prom Week*, but hoped the use of multiple characters and templates would make any given instantiation vary significantly, reducing negative impact even if it was seen by the same player multiple times. Though we had many means of varying the content of the dialogue, there were questions surrounding how best to make use of them. How effectively did use of the random template make an otherwise static line of dialogue feel dynamic? Were the simplistic character-specific locutions actually making a discernible difference when two different sets of characters engaged in the same instantiation? Intuitively, making frequent use of the SFDB seemed like a powerful use of the system, but were there certain situations where referencing character history was more powerful than others? Having the answers to these questions would have helped focus our authoring effort substantially, giving us the knowledge needed to better write for this new domain of retargetable social exchanges.

To find answers to these questions, data mining was used to identify the types of instantiations that were being repeated during single runs through the game, and the frequency across these runs with which they were seen. These instantiations were identified by searching the 109,984 playtrace files generated by players between the initial release of *Prom Week* to November 22nd, 2013. *Prom Week* is split into several campaigns, which in turn are made up of a number of levels. A new playtrace is generated at the completion of every level, and contains all of the player actions taken in that level and all prior levels in that campaign.

Due to this cumulative nature of the playtrace files, only files that marked the end of a campaign were considered, or 15,967 unique playtraces. Each playtrace is an xml file that catalogues all user moves, and contains all requisite information to be able to recreate the social state the player would have experienced. For every instantiation that was written in the game, we checked to see how many traces contained the instantiation multiple times. Although the results were interesting – some instantiations were never seen more than once in a single playthrough while others were frequently seen multiple times – it largely just confirmed our suspicious that the technique of retargeting can and does lead to repetition. The questions that we actually wanted to answer involved discovering the narrative quality of the stories that contained repetition, and data mining alone was insufficient to help us see the answers.

We developed the technique of story sampling in part to discover the effects of repetition on the quality of the stories produced. Six instantiations that appeared multiple times within a single playtrace were selected for analysis. These six instantiations possessed a range of the types of locutions used; half of them used the SFDB, which we hypothesized would make instantiation repetition feel the least egregious. Of the remaining three, two made heavy use of the random locution, and the last only made use of the pragmatic templates and character specific locutions, which we predicted would have the worst results. For each of the six selected instantiations, three play trace files were found in which the instantiation appeared two or more times. These playtrace files were then run through a custom-built transcript generator, which generated the dialogue that the user playing the game would have actually seen. The transcripts were then passed off to the lead author of the game and another game author along with a rubric; using the rubric, the two authors were asked to read all of the 18 transcripts and give each a rating from zero to three. A rating of three indicated that the repetition was either completely unnoticeable, or it was noticeable but it actually enhanced the quality of the transcript. A rating of two meant that the repetition, though noticeable, felt different enough during each occurrence (either due to variations in the dialogue or to the differing social contexts in which the two instantiations occurred) that it didn't feel completely out of place. A rating of one meant that the repetition was highly noticeable, in spite of text or contextual variations, and a rating of zero meant that the repetition was simply unacceptable and detracted from the quality of the story. The authors did this rating independently of each other, and then reconvened when finished to discuss their results.

The raters agreed precisely on twelve of the eighteen transcripts, and were always within a difference of one for the other six. The twelve with highest correlation of agreement contained a sampling of all four rankings; three transcripts were rated with 0s, five with 1s, three with 2s, and one with a 3. Although the raters nearly agreed with each other on every transcript, looking at the transcripts with perfect agreement is important because they epitomize characteristics that the authors identify as desirable and undesirable for the stories the system creates. Some of the findings were to be expected; zeros often came from playtraces in which it appeared the player was 'grinding' a particular action, repeating it again and again with the same group of characters, and thus not providing the game with material to create new interesting contextual cornerstones. However, other discoveries provided new views into what makes repetition more or less acceptable in *Prom Week*.

One of the greatest determining factors was simply time elapsed between the two occurrences of the same instantiation. Even in situations where the exact same text was produced, if enough other dialogue had transpired between the two instances, the authors were inclined to give the transcript a rating of a 1 or 2. Several other insights were gained with regards to use of references to the SFDB. Although, as hypothesized, several of the highest rated transcripts made use of the SFDB, we were surprised to discover that some of the lowest rated ones did as well; contrary to our initial authoring beliefs, the SFDB was not a cure-all. This enabled us to find common patterns of successful, and less successful, SFDB use.

The best uses were when characters specifically referenced an event during the second instantiation that had happened after the first; this felt good because it showcased how the characters, and consequently the system, are reacting to new story beats that had been introduced to the world, even if they were reacting to them with largely recycled text. Surprisingly, the repetition of this recycled text was identified by the authors to be pleasurably comical, though admittedly when seen too frequently began to feel artificial. There were transcripts where the exact same SFDB reference was pulled in both of the instantiations, though even these were discovered to not be ubiquitously problematic; the SFDB stores two types of references: actions done by the player during gameplay as previously described, and 'backstory' references, events written during the authoring process which prepopulate the SFDB when the game begins. Repetition of backstory references was generally much more jarring than repetition of a reference to something the player actively made happen. The latter felt like the entire student body was abuzz with the characters', and consequently the player's, latest exploits; the former felt like they were at a loss of things to say and were dredging up irrelevant facts from the past.

In addition to enabling the authors to recognize differences between acceptable and less desirable instances of repetition, story sampling led to design insights that could be used to minimize the latter. Random locutions, in their current implementation, can only change a single line of text; this makes it impossible for future lines of text in the same instantiation to reference the text snippet the random locution selected. However, this look into transcripts showed that the random locution went a long way towards reducing the disruption of repeated text. If randomness could encompass changing multiple lines, it would become an even more effective tool for variation. The system also does not keep track of how many times a specific option in a random locution is used; prioritizing random options (or applicable SFDB references) that have not yet been seen would enhance their effectiveness. Moreover, frequently some of the random options are sillier than others; this could be a useful moment to help let characters' personalities shine by having goofier characters be naturally more inclined to use the weirder random options, while more conservative characters would tend to stick to normal utterances (an effect which could be proceduralized by ordering the random texts from least to most unexpected). We discovered that character-specific mix-ins were very useful in lessening repetition as well; they take linear authoring effort, but deeply enhance characters' expressivity. Adding more types of character specific locutions, and more options for each locution, seems like an easy win.

Another design insight is to author instantiations in which characters acknowledge the repetition. If characters continue to refer to the same SFDB event, it could trigger a new class of

instantiation in which characters react to the fact that the same situation keeps appearing, which could help retain the inherent humor in repeated references to the SFDB. A similar technique could be employed when players continue to grind the exact same social exchange between characters: instead of playing the same social exchange for the nth time, the system could redirect to a "persistent" social exchange, where the character is called out on their repetitious behavior (e.g., "stop flirting with me!").

Though these results were generated with the benefit of thousands of playtraces, the fact that ultimately only eighteen playtraces were analyzed in depth implies that a similar story sampling process could be carried out with a smaller pool of logs as well, perhaps using playtraces generated during a game's beta period. This would allow the insights from story sampling to inform the game design and authoring process before release.

4. CONCLUSIONS AND FUTURE WORK

This paper presented the technique of story sampling, in which realized story beats generated by players of an interactive story are analyzed to provide views into the shape of the narrative experience that data mining alone does not provide. An example use case of story sampling was presented using playtrace data from the interactive narrative *Prom Week*, to provide an understanding of the types of experiences the game created that could not have been ascertained at authoring time. This naturally led to ideas for improved game design and authoring efforts to ensure that the dynamic narrative content generated by game and player is in line with authorial goals.

Story sampling proved effective for understanding the shape of narrative experiences generated by *Prom Week*, but there remain many promising directions for future work. Implementing the new features of *CiF* and *Prom Week* inspired by this evaluation, and then employing further story sampling on the revised systems, would further verify the efficacy of story sampling. Though *Prom Week*'s story beats took the form of character dialogue in instantiations, it would be illuminating to see if story sampling is as effective in domains where story beats took different forms, such as narration or animation. *Prom Week* was also a finished product; applying story sampling in an environment in which the interactive narrative is still being developed, such as a beta, could reveal how useful story sampling is in shaping authorial direction when integrated into the active game design process. And finally, the four step process of story sampling is potentially generalizable to domains other than interactive stories. For example, it would be interesting to see the story beat equivalent of the play traces used to generate the Halo heatmaps (perhaps a short video from the player's perspective leading to the moment of death). The heatmap showed that players were dying; story sampling in this case might better illuminate why they were dying.

5. REFERENCES

[1] Bernstein, M. 2007. Storyspace and the Making of Grammatron.

[2] Bernstein, M. 2013. Tinderbox.

[3] Drachen, A. and Sifa, R. 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. *IEEE Conference on Computational Intelligence and Games (CIG)*. (2012), 163–170.

[4] Grappiolo, C. et al. Using Reinforcement Learning and Artificial Evolution for the Detection of Group Identities in Complex Adaptive Artificial Societies. *itu.dk*.

[5] Kim, J.H. et al. 2008. Tracking Real-Time User Experience (TRUE): A comprehensive instrumentation solution for complex systems. (2008), 443–451.

[6] Klimas, C. 2013. Twine.

[7] Li, H. et al. 2013. Discovery of Player Strategies in a Serious Game. *First AAAI Conference on Human Computation and Crowdsourcing*. (2013).

[8] McCoy, J. et al. 2010. Authoring Game-based Interactive Narrative using Social Games and Comme il Faut. *Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate* (Providence, Rhode Island, 2010).

[9] McCoy, J. et al. 2013. Prom Week : Designing past the game / story dilemma. *Proceedings of Foundations of Digital Games (FDG 2013)* (2013).

[10] McKee, R. 1997. *Story: Substance, Structure, Style and the Principles of Screenwriting*. ReganBooks.

[11] Medler, B. et al. 2011. Data cracker: developing a visual game analytic tool for analyzing online gameplay. *Proc. CHI 2011* (2011), 2365–2374.

[12] Moura, D. et al. 2011. Visualizing and understanding players' behavior in video games: discovering patterns and supporting aggregation and comparison. *ACM SIGGRAPH 2011 Game Papers*. (2011).

[13] Sifa, R. et al. 2013. Behavior Evolution in Tomb Raider Underworld. *Computational Intelligence in Games (CIG)*. (2013), 1–8.

[14] Tak, S. and Ko, H.-S. 2005. A physically-based motion retargeting filter. *ACM Transactions on Graphics*.

[15] Thompson, C. 2007. Halo 3: How Microsoft labs invented a new science of play. *Wired Magazine*. 15.9, (2007).

[16] Weber, B. et al. 2011. Using data mining to model player experience. *FDG Workshop on Evaluating Player Experience in Games* (Bordeaux, 2011).