

# Shepherd: An Incremental Story Sifting-Based Drama Manager

Sage Deo, Jonathan Chung, Joshua McCoy

University of California, Davis  
sagevdeo@gmail.com, jonathanchung1113@gmail.com, jamccoy@ucdavis.edu

## Abstract

Incremental story sifters analyze an in-progress simulation to extract interesting narrative content or find a set of events that have the *potential* to become more narratively interesting if followed up in a certain way. There has been some investigation on the potential for an incremental story sifter to suggest future narrative events to human authors, but the technique of guiding a simulation using story sifting, without any human interference, remains completely unexplored. Thus, we present Shepherd, a drama manager powered by incremental story sifting, which guides otherwise completely autonomous characters toward making narratively interesting choices.

## Introduction

In the field of interactive emergent narrative, there is a distinction made between a chronicle of simulated events and a narrative. The key element that transforms a chronicle into a narrative is curation, the process of cutting out narratively irrelevant content from a chronicle to create a narrative. However, this raises an important question: how do we decide which parts of a chronicle constitute a narrative?

Oftentimes, as in the case of *Bad News* (Samuel et al. 2016), humans are able to intuitively find small sequences of narratively interesting content, and then assemble those sequences into a larger narrative. This process has been termed “story sifting.” To aid in automatic narrative curation, automatic story sifters such as Felt (Kreminski, Dickinson, and Wardrip-Fruin 2019) and Sheldon (Ryan 2018) were created. These automatic story sifters accept human-authored “story templates,” which are high-level descriptions of small sequences of narratively interesting material. Given a chronicle of simulated events and a set of story templates, automatic story sifters can identify themes such as “violation of hospitality” or “repeated betrayals,” which can later be assembled into a narrative.

Later, the notion of “incremental story sifting” was implemented in Winnow (Kreminski, Dickinson, and Mateas 2021), the successor language to Felt. Winnow is capable of identifying partially-fulfilled story templates in a simulation currently in progress, which the authors note could potentially be used for foreshadowing future plot developments, i.e. the completion of the partially-filled template in

question. However, this also opens up the possibility of using Winnow, or any other incremental story sifter, as a sort of drama manager to guide a simulation in progress towards making choices that progress any partially-filled story templates it detects. This was explored to some degree by *Loose Ends* (Kreminski et al. 2022), a mixed-initiative authoring system which uses Winnow to suggest plot developments for a human to choose between. However, this technique has not yet been explored as part of a fully autonomous simulation, without any human guidance.

In this work, we contribute the Shepherd system, the first (to our knowledge) instance of a story sifter being leveraged for drama management in a simulation with no human input. Shepherd differs from other story sifters due to the fact that it sifts alongside the generator, allowing our system to create focused, curated narratives using the variance of a chaotic generator. It nudges the course of the narrative rather than outright demanding that certain events take place, allowing our system to follow story structures in interesting ways while still retaining a degree of control and coherence. The creativity and cohesion of the stories our system will generate will be proof that this method of incremental story sifting alongside generation is a viable and ripe path to pursue.

In addition to these contributions, we provide a playable online version of a sample Shepherd project<sup>1</sup>, as well as its source code<sup>2</sup>.

## Related Work

This work is heavily informed by *Curating Simulated Storyworlds* (Ryan 2018) and its philosophy on story sifting and emergent narrative, especially as that philosophy is instantiated in the theatrical performance piece *Bad News*. We also directly use the Winnow incremental story sifter in our implementation of this system. Our use of story sifting as a drama manager, to guide a simulation in progress, is a simulationist extension of its use in the *Loose Ends* system. Where *Loose Ends* is a “strong story” system, in which a story sifter acts to guide a human author with ultimate control over the plot, our proposed system takes a “strong autonomy” approach, where the story sifter guides the characters, who otherwise make their own decisions.

<sup>1</sup><https://l00tkek.github.io/shepherd/>

<sup>2</sup><https://github.com/L00tkek/shepherd>

Guided by the work of Max Kreminski, Noah Wardrip-Fruin, and Michael Mateas in *Authoring For Story Sifters* (Kreminski, Wardrip-Fruin, and Mateas 2022), we intend to address the most glaring issue with interactive emergent narrative: “the dissolution of the player-perceived story into a structureless mess”. While Shepherd characters make their own decisions, they are guided by the drama manager to advance and complete story arcs. By sifting simultaneously with generation, Shepherd provides structure to the story itself rather than looking for narratives post-generation as the wizard of *Bad News* does. Although this sacrifices some randomness and chaos during the generation phase, the corresponding reward of coherence is far more impactful. In addition, because Shepherd remains a “strong autonomy” system, the characters continue to make decisions and create stories that surprise. As a result, Shepherd will generate stories that deviate from the established story templates which players will narrativize themselves as they do in *Dwarf Fortress* or *The Sims*.

## Technical Description

The basic building blocks of a Shepherd world are time, characters, actions, traits, and templates. The interactions between all of these pieces, as well as the drama manager, form the basis for rich emergent stories.

Consider a hypothetical Shepherd story focusing on the cartoon character Garfield. Suppose that over the course of this story, Garfield is repeatedly cruel to his housemate, the dog Odie, and that at the end of the story, Odie is kind to Garfield. We will use this story to illustrate how each building block of the Shepherd system helps create narratively interesting stories.

### Time

In Shepherd, time is divided into a series of discrete time steps. For each time step, each character in the simulation may perform one action. For ease of computation, all actions are internally represented as taking place in sequence. However, readers may construct narratives in which actions within the same time step happen in parallel. This openness to interpretation helps facilitate narrativization.

### Characters

Characters in Shepherd are mostly-autonomous agents. Every time step, each character generates a utility score for each possible action they can take, based on that character’s traits. The utility score is combined with the drama manager’s score for that action to produce an overall score. Then, the character makes a weighted random choice between all possible actions, with each action’s overall score used as its weight.

In the hypothetical story given above, there are two major characters: Garfield and Odie. Others may exist in the simulation, but we will focus on these two. The simulation is primarily moved forward by these characters taking actions, and the hypothetical story given above emerges from these interactions.

```
Camille: friendly, gourmet, royal
Felix: friendly, pious
Flora: pious, flirty
Vroronin: flirty, gourmet
Golden: pious, friendly
Ysemay: unfriendly, pious
Betsy: flirty, friendly
```

Figure 1: An excerpt from the Shepherd UI, showing each character’s name and traits.

This method of modeling character decisions allows for character choices to follow consistent trends, which is good for allowing the audience to understand each character. However, it also allows characters to infrequently surprise the audience, creating narrative intrigue.

### Actions

An action is an event in the story world that can be triggered by some character (the actor). Some actions are monadic, and have only an actor; however, other actions are dyadic, and involve a separate character who is not the actor (the target). Each character may only serve as the actor for one action per time step, but each character can serve as the target for arbitrarily many actions per time step.

Each action also has a set of tags that provide higher-level information about each event: some examples of tags are “friendly,” “food,” and “violent.” Tags mainly interact with the rest of the system in two ways. First, when evaluating an action (via their traits), characters consider that action’s tags when determining its utility score. Second, human-authored story templates can match actions based on tags. This allows a single line in a pattern to represent a wide variety of actions which are all narratively equivalent in the context of that pattern.

In our example story, we know of at least two actions, both of which are dyadic. One action is the action Garfield takes to be cruel to Odie: let’s decide that this action is called `kick`, and is tagged as “cruel.” Similarly, the action Odie takes toward Garfield should be tagged “kind”; we can call the action `giveLasagnaTo`. However, we could just as easily call the cruel action `rob` and the kind action `giveCandyTo`, and the example story would still take the same general shape (though it would be less appropriate to the setting of *Garfield*). This shows one of the tagging system’s strengths: it allows, but does not force, both authoring and character decision-making to consider *categories* of actions, based on how those actions affect other characters or how they are felt within the narrative.

### Traits

Traits are the main agents of character personality modeling, and provide the basis for utility-based character decision-making. Each character is assigned a consistent number of (currently, two) traits at character generation, and each trait has its own utility function. A character’s utility score for a given action is the sum of the utility scores for each of the

character’s traits evaluated on that action. At a high level, this means that each character is likely to act in ways that are consistent with each of their traits: for example, a character who is friendly (high utility for actions tagged “friendly”) and a gourmet (high utility for actions tagged “food”) is very likely to want to share a meal with some other character, because sharing a meal is tagged as both “friendly” and as “food.”

Like tags, traits can also be used as conditions when writing templates. Thus, one character’s traits may influence other characters’ behavior via the drama manager.

Consider again our hypothetical story. In the *Garfield* comics, Garfield is often cruel to other characters. We can represent this using a trait, which we will call “unkind,” and characters with this trait should receive increased utility from (and thus be more likely to take) actions tagged “cruel.” To represent Garfield’s specific penchant for cruelty towards Odie, we could create two traits: “Garfield” and “Odie”, and cause characters with the “Garfield” trait to receive utility from taking “cruel” actions towards characters with the “Odie” trait. Based on these traits, it is very likely that, at any given time step, Garfield will choose to be cruel to Odie, as in our example story.

### Templates and the Drama Manager

Each template represents a sequence of narratively interesting events, such as a “drunken brawl” or “failed flirtation.” Templates are expressed as patterns in the Winnow language. Shepherd maintains a list of possible, partially filled, and complete templates; this list is updated each time an action is taken. Furthermore, when the drama manager evaluates an action, it checks to see if that action would advance any of its tracked story templates. Actions that advance many story templates are valued more highly by the drama manager, as are actions that advance story templates which are close to completion. This makes it more likely that characters take actions that are narratively interesting and which work towards resolving open plot threads.

The drama manager provides the last piece of our example story. Garfield’s traits make it very likely that he will be cruel to Odie regardless of narrative context, but an author can use templates to recreate the behavior of Odie being kind to Garfield *after* Garfield was cruel to Odie, by writing a template describing that exact situation, such as fig. 2. By loading such a template into Shepherd, an author can make it more likely that both Garfield and Odie will behave in ways that recreate our example story.

### Analysis

Shepherd successfully straddles the line between structure and chaos in its generation. The following example of one of Shepherd’s outputs demonstrates its ability to produce comparable amounts of variability to other interactive emergent narrative systems while simultaneously providing recognizable structures and coherence. Figure 1 shows the traits of the characters in this specific simulation. Just as in *Bad News*, refreshing or closing this tab destroys this particular simulation, granting it a certain ethereality; although, of

```
(pattern exampleStory
  (event ?e1 where tag:cruel,
    actor: ?c1, target: ?c2)
  (event ?e2 where tag:cruel,
    actor: ?c1, target: ?c2)
  (event ?e3 where tag:cruel,
    actor: ?c1, target: ?c2)
  (event ?e4 where tag:kind,
    actor: ?c2, target: ?c1))
```

Figure 2: A Winnow pattern corresponding to a template for the example story.

Camille and Flora share a meal.

Figure 3: An example of an action choice influenced by Camille’s “friendly” and “gourmet” traits.

course this simulation will enjoy greater longevity due to its presence in this paper.

Because the only way for the player to understand and get a sense for the personality of a character is through their actions and the traits, it becomes imperative that the characters act according to their traits. Shepherd succeeds here as the actions of the characters do line up with the traits that they are assigned. Figure 3 is the very first line of this simulation which shows Camille’s tendency to choose more actions tagged “eat.”

But characters don’t necessarily have to follow their traits. They deviate frequently from their traits, doing things randomly or to fulfill set story templates. However, the display of the traits themselves may alter the perception of the player and how they may narrativize events. For example, in fig. 4 Betsy randomly chooses the “violent” and “unfriendly” action of biting Camille. This runs contradictory to the established “friendly” trait for Betsy, but due in part to Camille’s reaction and to Betsy’s “flirty” trait the event can be narrativized by players in its new context. Thus, the traits of characters in Shepherd do a great deal to color the perceptions of their actions.

Figure 5 demonstrates the power of incremental story sifting alongside generation. Shepherd successfully nudges characters to pick actions that advance or complete story arcs, which allows for more satisfying consequences and results for actions and alleviates the “structureless mess.” Shepherd can handle and track any number of story templates which allows for the tracking and conclusion of mul-

Betsy bites Camille!  
Camille flirts with Betsy.

Figure 4: An example of a surprising choice arising from Shepherd’s algorithm for character decision-making, which players can narrativize in interesting ways.

```
Felix murmurs a prayer for Golden.  
Golden murmurs a prayer for Felix.  
Golden suddenly shouts at Felix!
```

Figure 5: An example of how Shepherd’s story templates result in more structured story arcs. This series of actions took place over multiple time steps.

multiple arcs at a time. The successful completion of story arcs with clear beginning, middle, and ends provides catharsis and a sense of logical progression.

## Conclusion

We have introduced Shepherd, an incremental story sifting-based drama manager that guides simulations towards producing interesting narratives. Shepherd represents a completely new application of story sifting which is ripe for further exploration and evaluation. The system is very easy to author for, is extremely scalable and extensible, and results in the emergence of surprising and interesting character behavior from otherwise unstructured simulation, while still leaving room for players to narrativize. We hope that, in the future, this new paradigm can aid in the creation of interactive digital narratives that offer the usual pleasures of emergent narrative while minimizing some of the associated pains.

## Future Work

There are many possible ways to expand on the work presented here. The system would greatly benefit from having more content, such as more traits, actions, and templates. Furthermore, Shepherd itself can be improved in many ways: we hope to expand the state-tracking capabilities of the system so that, for example, character relationships can be better modeled and act as narrative material. We also hope to create authoring tools to make writing for Shepherd easier, especially for authors who may not be well-versed in programming.

There is also work to be done to properly evaluate Shepherd, both from the perspective of authors writing for Shepherd and players experiencing the resulting narratives. From the author perspective, it may be wise to conduct a user study, especially once the aforementioned authoring tools have been implemented. From the player perspective, another user study might help evaluate the interestingness of simulations guided by Shepherd. To fully evaluate Shepherd in production, we also hope to mount it in a fully realized narrative game, with a larger volume of authored content, graphical output and increased player interactivity.

Lastly, one shortcoming of Shepherd is that, though it generates character actions that follow predetermined story templates, the arc of each story may be hard for players to discern. This is true of most or all simulation-based systems, but Shepherd is unique in that it constantly tracks the progress of each story arc; thus, it is uniquely positioned to make this information more accessible to players. One potential way of doing so is inspired by *Loose Ends*: in that

system, each action that progresses a story goal is associated with some text that situates that action in the context of the story goal in question. A similar approach could be integrated with Shepherd to contextualize each action with respect to all story templates progressed by that action. User interface changes may also help make story arcs more apparent by drawing a better continuity between each character’s actions, or even allowing players to see only the parts of the simulation that pertain to a specific character.

## References

- Kreminski, M.; Dickinson, M.; and Mateas, M. 2021. Winnow: A Domain-Specific Language for Incremental Story Sifting. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 17(1): 156–163.
- Kreminski, M.; Dickinson, M.; and Wardrip-Fruin, N. 2019. Felt: A Simple Story Sifter. In Cardona-Rivera, R. E.; Sullivan, A.; and Young, R. M., eds., *Interactive Storytelling*, 267–281. Cham: Springer International Publishing. ISBN 978-3-030-33894-7.
- Kreminski, M.; Dickinson, M.; Wardrip-Fruin, N.; and Mateas, M. 2022. Loose Ends: A Mixed-Initiative Creative Interface for Playful Storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 18(1): 120–128.
- Kreminski, M.; Wardrip-Fruin, N.; and Mateas, M. 2022. *Authoring for Story Sifters*, 207–220. Cham: Springer International Publishing. ISBN 978-3-031-05214-9.
- Ryan, J. 2018. *Curating Simulated Storyworlds*. Ph.D. thesis, UC Santa Cruz.
- Samuel, B.; Ryan, J.; Summerville, A. J.; Mateas, M.; and Wardrip-Fruin, N. 2016. Bad News: An Experiment in Computationally Assisted Performance. In Nack, F.; and Gordon, A. S., eds., *Interactive Storytelling*, 108–120. Cham: Springer International Publishing. ISBN 978-3-319-48279-8.