Data Visualizations For Complex Computational Narratives

Jacob Garbe, Noah Wardrip-Fruin, and Michael Mateas

UC Santa Cruz, Santa Cruz CA 95060, USA,
jgarbe@ucsc.edu,
https://games.soe.ucsc.edu/eis

Abstract. Computational storytelling, in both games and the larger field of interactive media, offers a wide range of new narratological forms that are otherwise impractical–or impossible–to create. A variety of systems push at these possibilities, but largely the potential narratological complexity outpaces our human ability to effectively author for them. Because of this, authors are limited not so much by the technology, but the daunting task of content creation that leverages the full affordances of the system. We posit a critical component of authoring for these systems involves visualizations of the work's emerging shape and dynamic state, and offer an illustrative case study of a viz for a dynamic choice-based narrative system currently under development.

Keywords: interactive narrative, data visualization

1 Introduction

Computer-based narrative has posed the challenge of superhuman structure, even in the early days of hyperfiction. As remarked by Mark Amerika, author of Grammatron (1997): "creating complex hypertext structures for the web is a nightmare because, after a certain point, one cannot visualize a cognitive mapping structure for a webwork that has literally thousands of screens and links" [1]. This complexity goes hand-in-hand with authorial overhead that–for the most part–is still largely shouldered by authors without procedural assistance. Because of this, unintended consequences of content authoring, such as uneven content coverage, unplanned state changes, and mis-directed narrative pathways can go unnoticed unless laboriously checked by hand. This labor requirement unnecessarily shackles narratives in both independent and commercially shipped interactive media, and discourages creative investment in more complex systems. In order to push the envelope of interactive narrative, therefore, it'd be fruitful to address the overarching problem of effective authorship, and work to create tools that can surface the underlying narrative structures to authors effectively.

2 Prior Work

A large subset of interactive narratives, namely hyperfictions / choice-based narratives and tree-based dialogues in games, have been represented via directed graphs. This type of visualization strategy has worked since the early days of hyperfiction, with tools such as HyperCard, Aquanet [8] and Storyspace [2], and persists through today in hypertext authoring environments such as Twine [5] and Tinderbox [3]. Professional game tools such as Articy: Draft [6] also make use of this strategy, incorporating state information as annotation to the graph nodes, while modding resources such as Bioware's Aurora/Odyssey/Eclipse engine [10] still use dialog tree models as nested lists.

Most of these viz tools display static connections between lexia without taking state-conditional content selection into account - meaning graphs either display all possible connections that could happen (but don't help you understand under what conditions a content sequence will actually happen) or fail to highlight important connections. As such, while more informative than the still-widespread use of spreadsheets, creators are still limited by unpredictable side-effects of state-based choice content. Additionally, the multi-linearity of stories can quickly overwhelm the writer, especially if (as with Twine and other tools) there aren't automatic layout algorithms to handle content addition appropriately, forcing the author to manually re-position nodes.

Some tools tackle the state problem, such as the Inform 7 Skein tool [9]. In order to account for the added difficulty of state, this class of visualizations switch to a playthrough model, aggregating their path through content and displaying nodes that are visited. In the case of Inform 7, however, it is infeasible to explore every "choice" (action) at every narrative point, as the graph explosion would pollute the visualization with unhelpful states. As such, authors are forced to manually specify traversal paths.

We have also previously developed procedural, interactive visualization tools [4] to assist with content authorship for combinatorial narratives, specifically *The Ice-Bound Concordance* [7]. An interactive demo of said visualizations can be found at http://www.ice-bound.com/viz/demo.

There is still more to explore in this space, which in turn can enable more variegated forms of computational narrative. We'd like to present a procedural visualization used to help enable such narratives, by providing verification and exposing structural problems that might otherwise remain hidden.

3 Case Study: StoryAssembler

StoryAssembler is an ongoing project built around a model of procedurally assembled choice-based narrative. The general architecture of the system is as follows:

1. An unordered "wishlist" of desired story states is passed to the system
2. The content library consists of lexia, choice labels, and destination links
3. Lexia can modify the story state when visited by the player
4. Choice labels can be statically assigned to destination links, or dynamically chosen
5. Which choices are paired with lexia can be static, or dynamic based on how they modify the state

6. The system greedily searches for lexia which will satisfy wishlist items, and greedily populates choices for said lexia with options that lead to lexia that satisfy wishlist items

What defines a "good" narrative experience for this work? If the system is "firing on all cylinders", it should handle different "wishlists" (within reason) by assembling new choice structures. It should also re-purpose content at different points of the structure to satisfy wishlist items. Last, it should select content from a bank of options, not a static 1-to-1 matching that's pre-authored.

3.1 Challenges and Solutions

Authoring content for this system is difficult to verify by hand, due to emergent effects caused by the unordered nature of wishlists, and the greedy nature of the search. Content the author may intend to show up at a certain point may appear much earlier or later, due to unintentionally satisfying desired state conditions.

Additionally, the knee-jerk reaction to prescriptively restrict lexias with stringent pre-conditions undermines the core goal of the system: if there is only one possible point the content can appear, then it might as well be statically linked. Ideally, all content in these narratives should be capable of appearing in at least one other narrative position.

Without a tool, the only way to double-check content authoring is through laborious traversal of the choices, and given that they are also dynamically assembled, essentially the entire structure must be re-verified with each added lexia or choice, to insure it isn't showing up in an undesired spot due to unforeseen state conditions.

The visualization solution we created for this problem needed to show the assembled choice structure, at what point wishlist items were being fulfilled, and how content was being re-used. We decided to use the playthrough approach, using the same procedures called in-program to collect data. The resulting data was displayed as a directed graph. However, additional strategies were used to expose some of the underlying structure.

We created a subset of the narrative's state variables to establish whether a node in the structure could be considered structurally identical under different contexts. A good example is a short test segment where the main character is sitting in an airplane reading the newspaper. After an introduction (which fulfills an introductory wishlist item) she has a choice to read any of four articles, incrementing a state variable "articlesRead". When four articles are read, the last wishlist item is fulfilled and the segment ends.

If we set the visualization to not consider articlesRead as a differentiating state value, we get a flower-like structure with the central node as the recurring lexia (Fig. 1a). If we set articlesRead as a differentiating state value, however, we get a slightly different structure (Fig. 1b) where the recurring node, although identical in content, appears as a separate entry. However, they are still grouped together in a box, due to their shared content ID.

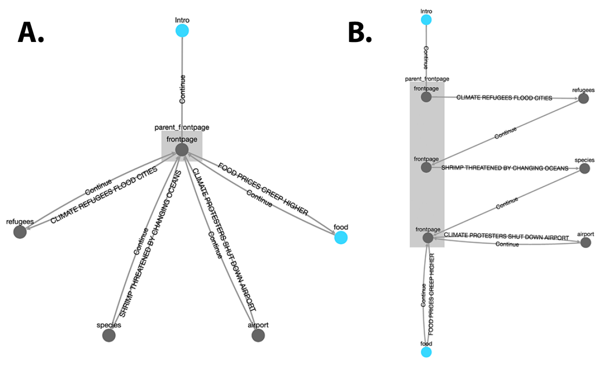

Figure 1. Screenshot from interactive visualization

Each node has a contextual menu that can be opened to display which nodes are valid for the system to display after the selected node, which are invalid, and the reason they aren't. Combining this with the state variable filters means we can inspect to see why a recurring node may not go where intended, or link to an unexpected place. We can also simply replay the traversal to that node. Last of all, if the node satisfied a wishlist item (denoted by the color blue) we can see which items it satisfied.

These features help quickly show when added content is changing the assembled choice structures in undesirable ways, and also shed light on what sort of content is still needed in order to satisfy wishlist items.

3.2 Interactive Demo

An interactive demo of the system can be found at https://games.soe.ucsc.edu/storyassembler.

4 Future Work

While a critical authoring tool for our system, this style of visualization could also be applied to other multi-linear branching narratives with JSON data, given a conversion parser is written. Particular low-hanging fruit would be Twine or Choice of Games narratives, as their data is already in JSON format. This could potentially provide a rich field of visualizations that could further scholarship with these narrative forms in the future.

5 Conclusion

The field of interactive storytelling media has seen incredible progress, driven by ever-improving technology and an ever-widening audience. We believe the narratives of these work can be pushed to concomitantly deeper and more complex forms, through the assistance of procedural tools crafted to present state and context information to authors as they create.

References

1. Bernstein, M.: Storyspace and the Making of Grammatron. http://www.eastgate.com/ storyspace/writing/Amerika.html (2007)
2. Bernstein, M.: Storyspace: hypertext and the process of writing. Hypertext/hypermedia handbook 529-533, (1991)
3. Bernstein, M.: Tinderbox. http://www.eastgate.com/Tinderbox (2013)
4. Garbe, J., Reed, A., Dickinson, M., Mateas, M., Wardrip-Fruin, N.: Author Assistance Visualizations for Ice-Bound, a Combinatorial Narrative. Foundations of Digital Games (2014)
5. Klimas, C.: Twine. http://twinery.org (2013)
6. Nevigo.: articy:draft [Software]. http://www.nevigo.com/en/articydraft/try/ (2016)
7. Reed, A., Garbe, J., Wardrip-Fruin, N., Mateas, M.: Ice-bound: Combining richly-realized story with expressive gameplay. Foundations of Digital Games (2014)
8. Shipman III, F.M., Marshall, C.C.: Spatial hypertext: an alternative to navigational and semantic links. ACM Computing Surveys (CSUR, 31(4es):14 (1999)
9. Reed, A.: Creating interactive fiction with Inform 7. Cengage Learning 367-370 (2010)
10. Bioware: Game Engines. https://en.wikipedia.org/wiki/BioWare. 2016.