Cast: A Context-Aware Collaborative Storytelling Platform

Gabriel Caniglia
Northwestern University
Evanston, IL 60201, USA
gcan@u.northwestern.edu

Abstract

The process of creating stories with others is highly engaging but labor-intensive, especially because satisfying narratives often require a central organizer to provide structure and scaffolds. To make this task easier for the organizer and the participants, we propose the use of context-awareness as a way to scaffold participant contributions in a collaborative storytelling experience. We have created Cast, a platform that allows an organizer to concisely define a story script that is opportunistically deployed to users through a context-aware mobile app. Leveraging the contexts of users to define their roles within a collaborative story provides them with scaffolds that promote the development of a compelling narrative, and automating this process prevents burdening the organizer. A pilot study demonstrated that stories developed with Cast are engaging yet low-effort for participants, showcasing the potential for context-awareness to make collaborative storytelling more accessible without sacrificing enjoyability.

Author Keywords

Context aware; collaborative computing; storytelling

CSS Concepts

• **Human-centered computing~Collaborative and social computing**

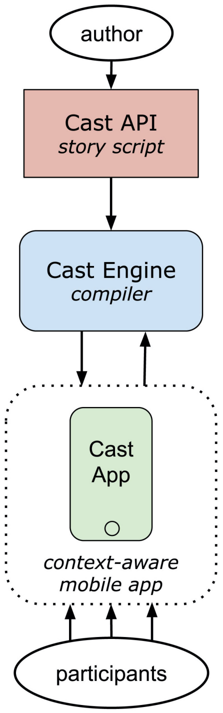


Figure 1: An author writes a story script using the Cast API. The Cast Engine uses the script and context-awareness to scaffold the participants as they develop the story together within the Cast App.

Introduction

Collaborative storytelling involves multiple people working together to develop a narrative. Compelling collaborative stories require significant effort from participants but are also highly rewarding for them [5]. Participation within a collaborative story is interactive, social, and immersive in ways that the passive consumption of stories, such as watching a movie, cannot match. For example, an increasingly popular type of collaborative storytelling is the "murder mystery party," where a group of people get together to act out a murder investigation. Each person is assigned a character persona, one person is secretly designated the murderer, and the others must collectively solve the crime as clues are slowly revealed.

These types of collaborative stories are incredibly fun and engaging for participants, but they are also the hardest to execute. They require a central organizer to provide *scaffolds* as a way to guide participant contributions and enforce a story structure [5]. These scaffolds include tasks such as planning the experience and creating and assigning roles, which take significant time and effort from central organizers. But without the scaffolds, collaborative stories often end up as disjointed and aimless, making them less engaging for participants and less satisfying as narratives [6]. And even with scaffolds, there is still a high barrier to entry for participants, as they need to convincingly act out their personas and collaborate effectively with others to ensure the story remains cohesive. These many obstacles mean that the most enjoyable collaborative stories need to be planned far in advance and cannot happen very often. Ideally, creating and participating within collaborative stories could be much easier, without sacrificing the quality of the experience for participants.

While existing digital approaches have made storytelling easier than or similarly engaging to the best collaborative stories, none have addressed the issue of scaffolding participants in a collaborative storytelling experience. Digital collaborative storytelling tools either forgo central organizers and scaffolds altogether [2], or they do little to ease the burden on central organizers, making their tasks just as hard. These approaches result in simple yet unengaging experiences and unsatisfying stories [6]. However, context-aware storytelling, such as a location-based experience where a participant uncovers a narrative as they walk around, is a form of digital storytelling that has been shown to be similarly engaging to robust collaborative stories [3]. While no existing solutions combine context-awareness with collaborative storytelling, doing so could provide an automated way to scaffold participants, reducing the burden on a central organizer while maintaining an engaging experience.

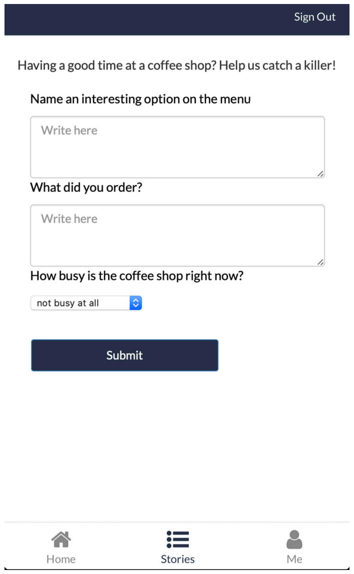
To demonstrate the potential for context-awareness to make collaborative storytelling easier for authors and participants without sacrificing quality, we present Cast, a context-aware platform for collaborative storytelling. Cast involves (1) the Cast API that allows authors to concisely define collaborative story "scripts," (2) the Cast context-aware mobile app that enables participation, and (3) the Cast engine that automatically converts story scripts into opportunistic, collaborative storytelling experiences (Figure 1). For example, if an author wants to create a murder mystery, they could use the Cast API to situate their murder mystery within a specific context, such as

```
let setting = 'at a cafe';
let notification = 'At a cafe?
    Take part in a murder
    mystery!';
```

Figure 2: The setting establishes the story's base context.

```
let question1 = {
  question: 'What did you
    order at the cafe?',
  responseType: 'text',
  responseData: 'order'
}
```

Figure 3: The questions help further define and distinguish each user's specific context.


Having a good time at a coffee shop? Help us catch a killer!

Name an interesting option on the menu

Write here

What did you order?

Write here

How busy is the coffee shop right now?

not busy at all

Figure 4: Users answer the pre-story questions before the story begins.

cafés. After defining character roles (such as murderer and innocents), the author can define "at a café" as the specific context they want participants to be within in order to participate in the story. Then, the author can create questions to better understand the contexts of participants, such as what they ordered at the café. Lastly, they can write narrator-driven prompts that leverage this contextual information to progress the story. Users can then participate within the story using the Cast app. For example, if the person cast as the murderer ordered a latte, the narrator would reveal this information to the other participants. The users could then send photos and messages; the murderer would try to skirt around the evidence and deceive the innocents, while the others would try to prove their innocence (Figure 8).

Cast enables a broad array of engaging collaborative stories while drastically reducing the time and effort required from an author. Instead of carefully building personas for each character, an author only needs to create context-defining questions that allow for participants' contexts to dynamically define their personas. And the author no longer needs to act as a central organizer: the Cast engine automatically assigns participants to a story when their contexts fulfill the script. Participation is also easy for users, as their personas are defined with their real-world contexts, making it easier for them to act within the story. Because the Cast app opportunistically identifies situations where users can participate in a story, collaborative stories no longer need to be planned in advance and can occur more often in everyday contexts. Lastly, stories made with Cast remain compelling and engaging: the author-defined narrator prompts scaffold the participants and encourage them to send messages and photos that progress the story and make the experience fun.

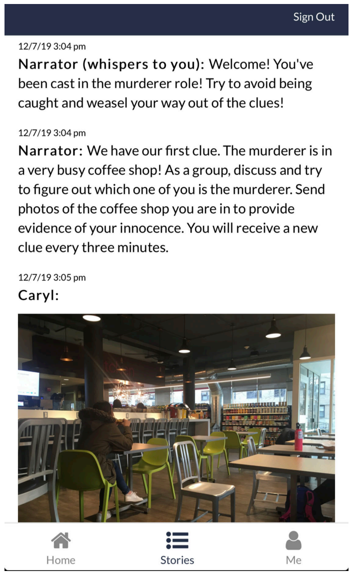
The main conceptual contribution of this work is the idea of *automating collaborative storytelling scaffolds through context-awareness*. Cast uses the engaging aspects of context-aware storytelling to generate scaffolds and remove the need for a central organizer, reducing the burden on the author while still providing a satisfying experience for participants. The main technical contribution of this work is *a storytelling API that abstracts context-aware and collaborative storytelling affordances*, enabling authors to easily implement an automated, scaffolded, and collective narrative. Cast therefore enables stories that are structured by an author, mediated by contextual affordances, and developed through the collaboration of multiple participants. Throughout the rest of the paper, we use the example of murder mystery to demonstrate Cast's implementation and feasibility. While future evaluations of Cast are necessary to understand whether it can make an author's role easier in practice, our pilot study does support the idea that context-awareness can make collaborative storytelling easy and engaging for participants.

Related Work

Existing digital approaches to collaborative storytelling have made the process easier, but they have done so by sacrificing the quality of the stories they facilitate. Some tools simplify the process by eliminating a central organizer, such as websites where users can write individual chapters of a larger story [2]. While easy to participate within, these stories are often disjointed without a person or system enforcing cohesion. Other systems involve human moderators to act as the

```
let char1 = {
  roleName: 'murderer',
  instruction: 'You're the
    murderer! Avoid getting
    caught by masking your
    context!',
  context: ['busy'],
  max: 1
}
```

Figure 5: Here, the murderer character is cast as the user in the busiest café.


Sign Out

12/7/19 3:04 pm
Narrator (whispers to you): Welcome! You've been cast in the murderer role! Try to avoid being caught and weasel your way out of the clues!

12/7/19 3:04 pm
Narrator: We have our first clue. The murderer is in a very busy coffee shop! As a group, discuss and try to figure out which one of you is the murderer. Send photos of the coffee shop you are in to provide evidence of your innocence. You will receive a new clue every three minutes.

12/7/19 3:05 pm
Caryl:

Figure 6: The user cast as the murderer receives a private "whisper" from the narrator, informing them of their role before the experience begins.

central organizer, but they provide no support tools for these moderators. Stories may be easier to create for participants, but they are no easier for the organizers to manage, which means they often still end up incohesive and unsatisfying [6].

Cast is also related to work in the areas of context-aware systems. We use context as defined by Abowd et al.: "any information that can be used to characterize the situation of an entity" [1]. Cast specifically builds off of the technical implementation of Cerebro [7], a context-aware social computing system that finds serendipitous moments for engaging in joint activities and facilitates interactions at a distance. Like Cast, it aims to make these moments easy to participate in through the use of automated coordination. However, Cerebro does not give the author of an experience a way to plan a narrative and define how participants are cast within it, making it a poor tool for scaffolding collaborative storytelling. Other attempts at context-aware storytelling, such as GEMS [3], have shown context to be successful at increasing participant engagement. However, GEMS only uses context to facilitate non-collaborative stories, not to scaffold participation in a collaborative story or to provide an API for authors to create their own context-aware stories.

Many forms of interactive storytelling, which usually involve an author defining every possible branch of a story, in the style of a "choose your own adventure" format, do provide authors with an API. These solutions, such as Inform, give authors a rich syntax for defining stories that have interactive elements [4], but they're still labor-intensive since they require defining every possible story permutation. And while participants get to play a more active role, their contribution is still limited to choosing between a finite set of predetermined options.

Cast, however, lets participants make their own original contributions to develop a story with details not prewritten by an author, making it easier for authors to scaffold thematic elements to maintain cohesiveness, while letting participants handle the details.

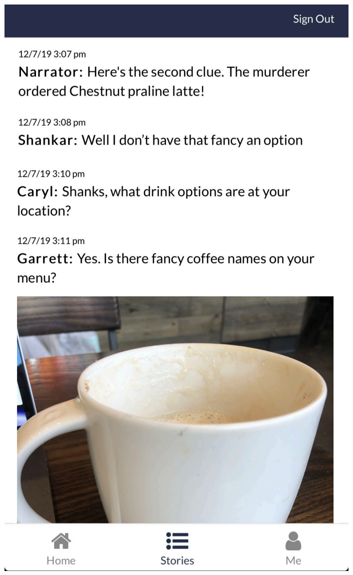
System Description

At the center of the Cast platform is the Cast API, which allows an author to define a story script in only a few lines of code. To create the murder mystery example, where participants in the context of a café use their surroundings to act out a storyline of slowly revealed clues that lead to discovering the murderer, an author defines a few parameters:

1. Setting: The required context for a participant to participate in the story, such as "at a café" (Figure 2).
2. Pre-Story Questions: When a participant joins a story, they must first answer these questions that allow the system to further understand their context, such as "What did you order at the café?" (Figures 3 and 4).
3. Characters: The roles within the story. The roles have casting conditions that are based on participant context, and include specific instructions only sent to those roles (Figures 5 and 6).
4. Prompts: These are sent by the narrator figure according to an author-defined timing function. They can be dynamically updated depending on participant context. For example, if the murderer ordered a latte, the prompt would reveal this information (Figures 7 and 8).

```
let prompt2 = {
  prompt: "Here's the second
    clue. The murderer
    ordered",
  info: 'order',
  timing: 200
}
```

Figure 7: This prompt will reveal what the participant cast as the murderer ordered.



12/7/19 3:07 pm
Narrator: Here's the second clue. The murderer ordered Chestnut praline latte!

12/7/19 3:08 pm
Shankar: Well I don't have that fancy an option

12/7/19 3:10 pm
Caryl: Shanks, what drink options are at your location?

12/7/19 3:11 pm
Garrett: Yes. Is there fancy coffee names on your menu?

Figure 8: The narrator sends a prompt with a new clue about the murderer, leading the users to interrogate each other to figure out who the murderer is.

Writing a short story script is the only action required from an author in order to create a collaborative story. All participant coordination is automated, and the dynamic contextual information from participants supplants the need for an author to spend time defining detailed character personas or story premises.

The Cast App monitors the user's location in the background, sending them a notification to participate within the story when their location matches the author-defined setting. The app provides a form for submitting pre-story questions (Figure 4) and a group chat for stories to take place within (Figure 6). The narrator figure can send "whispers" in this chat that only specific users see, or it can send messages to all users. Users can submit text or photos. At the conclusion of an experience, users can read through the entire story.

The Cast engine is a compiler that turns an author-defined script into a collaborative story. It first feeds the story's setting into Cerebro, which handles pinging users who are located within that setting. It then uses a callback structure that runs after a sufficient number of participants have submitted pre-story questions. User responses to these questions are stored within a database. The callback function then casts participants, using their answers to the pre-story questions and the casting conditions as defined by the author. Then the narrator prompts begin within the chat interface: first participants are notified of their roles, and then the prompts are released in succession based on the author-defined timing function. User responses to the pre-story questions are accessed throughout the experience for use within the dynamic prompts.

All of these automated functions act as scaffolds for the participants, ensuring easy participation and an engaging, cohesive story. Since the system is opportunistic, there are no planning and coordination costs for authors or participants. Casting conditions ensure participants have roles that fit their contexts, making them easy to act out, and dynamic prompts maintain narrative progression while adjusting to the specific contextual conditions of the story. The result is a simple experience for authors and participants that represents a complicated coordination system which would normally burden a central organizer.

Preliminary Study and Evaluation

To better understand if Cast can make collaborative storytelling easier for participants without sacrificing their enjoyment, we ran a pilot study using the Cast app and "murder mystery" experience. While this study provided valuable feedback about the participant experience, further work is necessary to understand if Cast also succeeds at lessening the burden on the central author.

The study had three participants who agreed to partake in a collaborative storytelling experience. Each participant was individually instructed to download the Cast app and visit a local coffee shop in order to participate in the café murder mystery. We ensured that they each visited a different coffee shop at around the same time. This triggered the experience to start: they all received and accepted the notification to participate on their phones, they filled out the pre-story questions, and they participated within the story itself. Afterward, they filled out a post-survey and we conducted brief follow-up interviews.

Overall, participants found the experience to be fun and engaging because of its context-aware and social qualities. They made abundant use of the Cast app's photo feature, sending many photos of their contexts to try and prove their innocence. For example, the system revealed that the participant cast as the murderer was in a busy café, so that participant took a carefully framed photo that made their café seem empty, helping keep their identity hidden. Afterward, they noted how this ability to use their context directly in the story made it fun to participate in. As participants joked with each other and sent accusations through their messages, they also demonstrated how Cast maintains the highly social qualities of successful collaborative stories. Afterward, one of the participants specifically mentioned enjoying these social aspects of the story, adding that it would be especially fun to experience with friends.

Participants also noted that the experience was engaging but low-effort. One participant said it was a fun experience "without investing a lot of time," and another specifically noted that "other murder mysteries [they've] been in required a lot more effort from the participants." The Cast story didn't require any preparation from the participants, as their assigned roles matched their current contexts and the system provided them with all necessary instructions. The feedback we received from participants aligned with one of our goals of providing an easier way to participate in engaging collaborative stories.

Discussion and Future Work

Our pilot study helped demonstrate that the contextually scaffolded approach implemented by the Cast platform can result in engaging, easy collaborative storytelling experiences for participants. While we do not yet have evidence that these experiences are also easier for authors to create, the effort required to organize and participate within a contextually scaffolded story is still far less than what is required for the central organizer(s) and participants within a traditional collaborative storytelling experience like a murder mystery party. Combining contextual scaffolds with collaboration shows promise as an approach for enabling a low-effort yet rewarding form of digital storytelling, allowing collaborative stories to occur more frequently within the everyday contexts of users. This context-aware approach can also bring benefits to more broadly defined collaborative systems, increasing opportunities for scaffolded cooperative engagement among users within other forms of entertainment or even productivity applications.

Future work includes better evaluations of the Cast system, especially through an evaluation of the authoring interface's effectiveness. We would also like to continue developing the Cast API to allow for the creation of broader, richer stories, especially for providing scaffolds within stories that span time and space.

Acknowledgments

We thank Ryan Louie, Haoqi Zhang, Sanfeng Wang, and the entire Design, Technology, and Research program for their continued guidance and support.

References

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. Towards a Better Understanding of Context and Context-Awareness. In *HUC 1999*. Lecture Notes in Computer Science, vol 1707. https://doi.org/10.1007/3-540-48157-5_29

[2] Jason Garber and Kevin Lawver. 2014. About Ficly. Retrieved January 5, 2020 from https://ficly.com/about/

[3] Jason Procyk and Carman Neustaedter. 2014. GEMS: the design and evaluation of a location-based storytelling game. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (CSCW ’14). Association for Computing Machinery, New York, NY, USA, 1156–1166. https://doi.org/10.1145/2468356.2468550

[4] Aaron Reed. 2010. *Creating interactive fiction with Inform 7.* Cengage Learning, Boston, MA.

[5] Scott Rettberg. 2005. All together now: collective knowledge, collective narratives, and architectures of participation. In *Proceedings of the 2005 Digital Arts and Culture Conference.* Copenhagen, DK.

[6] Sue Thomas and Bruce Mason. 2008. A million penguins research report. Institute of Creative Technologies. Leicester, UK.

[7] Jennie Werner and Allison Sun. 2018. Cerebro: A Platform for Opportunistic Collective Experiences. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI EA ’18). Association for Computing Machinery, New York, NY, USA, Paper SRC23, 1–6. https://doi.org/10.1145/3170427.3180298