Off the Beaten Plot: A Survey of Underexplored Story Generation Systems and Aspects

Nick Montfort[1,2,*,†], Jakob Kusnick[2,†]

[1]Massachusetts Institute of Technology, 77 Massachusetts Ave Room 14E-316, 02139 Cambridge, Mass., United States
[2]Center for Digital Narrative, University of Bergen, Langesgaten 1-3 NO-5020 Bergen, Norway

Abstract

We present a survey of story generation systems, seeking to answer two questions: First, given that story generators have made different contributions to our understanding of narrative, composition, and creativity, what are some underexplored aspects of the major systems? We examine what the scholarly record leaves unanswered regarding the different systems themselves and how these generators could be elaborated, combined, and modified to continue addressing the research questions that motivated them. Second, what story generation systems have typically been left out of other surveys and are not in the mainstream of research, but nevertheless may be interesting to consider alongside the more canonical systems? We consider, for instance, some story generators that are artistic and literary projects and some that are part of commercial software. Because this survey aims to fill these two gaps, we describe the most well-known story generators in little detail, referring the reader to existing surveys and similar resources.

Keywords

story generation, creative computing, computer history, miniatures

1. Introduction

Several others have recently surveyed story generation systems in papers [1, 2, 3] and books [4, 5]. Another view of these is presented in a recent anthology of output [6]. We were prompted to consider these systems as we search for candidates for reimplementation in our current research project, Narrative Nubs (see Subsection 1.1) [7]. This led us to revisit the history of story generation, highlighting underexplored aspects of well-known systems and bringing other systems into the discussion. With a focus on non-interactive and less interactive systems, we establish three categories: First, ● *canonical generators*, those referenced most often in research — we consider how to fruitfully investigate these further. Next, ■ *generators on the margins* that generate meaningful representations of event sequences but have seldom been discussed alongside canonical systems. Finally, we mention some ★ *distant outlier generators* that are intriguing, but difficult to fully bring into even an expanded view of story generation. We are not judging which generators are better or worse by placing them in these categories. We simply mean to reflect how these systems are situated in existing research and scholarship, using a (still biased) perspective we have as researchers undertaking a new project.

1.1. Narrative Nubs

Our ongoing research project, *Narrative Nubs* [7], involves developing "nubs" by reimplementing abstracted versions of existing story generation systems. This project was initiated by Montfort as a collaboration between MIT and the University of Bergen's Center for Digital Narrative. These nubs (as in "the nub of the argument") are miniatures [8] that preserve core concepts. We use Python 3 and follow a consistent coding style, producing encapsulated, reusable, and approachable components. We aim to

9th International Workshop on Computational Models of Narrative, June 8–10, 2026 - Madrid, Spain
*Corresponding author.
†These authors contributed equally.
nickm@nickm.com (N. Montfort); jakob.kusnick@uib.no (J. Kusnick)
https://nickm.com (N. Montfort); https://vokabelsalat.github.io (J. Kusnick)
0000-0001-7558-5160 (N. Montfort); 0000-0002-1653-6614 (J. Kusnick)

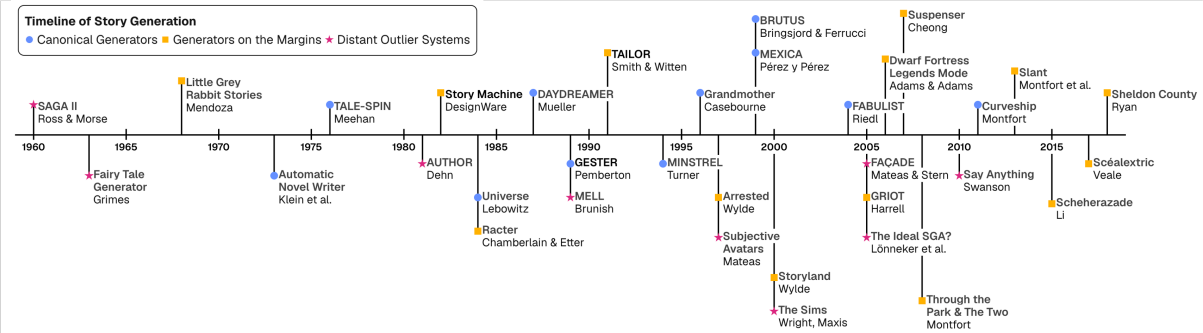

Figure 1: Timeline of story generators we survey in this paper.

computationally explore aspects of overlooked systems, ones often only described in the literature and in some cases known solely through their generated output. This is achieved through the re-enactment of historical approaches [9] and a strong emphasis on "learning by doing" [10]. We hope nubs will be used in research, teaching, and artistic practice, and will surface not only the underlying computational mechanics of the story generators but also the authors' assumptions and some aspects of the cultural, social, and historical contexts in which these systems were originally developed. Many researchers have already reimplemented historical story generators [8, 11, 12, 13, 14, 15]; these are inspirations for our set of consistently coded nubs. Four nubs have already been implemented: *Story Machine* [16], GESTER [17], TAILOR [18], and *Through the Park* [19]. To identify other candidate systems, we revisit the history of story generators to highlight overlooked aspects of some projects and discuss some that have eluded notice entirely. While we aim at a broader view of story generators, we admit, too, that some may inspire modern-day systems but not be good candidates for our Narrative Nubs project.

1.2. Our Understanding of Narrative and Story

Formally, we consider narrative as the representation of one or more events in sequence. Cognitively, it is a representation with narrativity, provoking us to imagine events in time. Rather than claim that a story generator must be structured in a certain way, we consider any software system that outputs narrative in either the formal or cognitive sense to be a story generator. We do not distinguish narrative from story, although one could note that stories have a point within some culture [20].

2. Canonical Generators

Some story generators are staples of survey articles and books (i.e., [1, 2, 3, 4]). These are typically plot generators that do not focus on surface text generation; exceptions will be noted. Many of these have been further developed or reimplemented. Nevertheless, some of their aspects are underexplored.

2.1. Novel Writer

Sheldon Klein's Automatic Novel Writer (1973) generates short stories (not novels) somewhat like murder mysteries [21]. The system introduces characters at an English country manor, then moves through flirtations and romantic encounters towards the homicide. The style and story world are consistent, although the text is rambling as events are decided randomly and presented using a pre-written set of templates. Novel Writer May be the first system with an underlying model of a story world. It shows that even a randomly selected plot with a few constraints can evoke a particular genre, raising the question of whether the same primitive-seeming method could generate narratives in different genres. Oddly, the murderer is revealed to the reader in the middle of the story, even though the system seems powerful enough to keep the story suspenseful. A nub could help explore remedying this and generating stories in other genres [22].

2.2. TALE-SPIN

TALE-SPIN (1976) by James R. Meehan is a pivotal system, even if its outputs seem belabored and simple [23]. Its complex simulations have multiple characters with goals, beliefs, and social awareness. TALE-SPIN influenced projects and has served as the basis for multiple reimplementations [8, 15, 24, 25, 12]. Discussions of the system have very often remarked on the "mis-spun" tales documented by Meehan. The implementation of the force of gravity as a character, for instance, led to:

"Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned." [23]

Reimplementations of the system, however, aim for proper output. A nub could provide insight into TALE-SPIN as it existed when Meehan completed his dissertation, but also the earlier versions that generated mistakes with amusing and even literary qualities.

2.3. UNIVERSE

The system UNIVERSE (1984) by Michael Lebowitz works to generate stories with consistent characters and evolving plots, as in soap operas [26]. An early planning-based generator, it produces episodic, open-ended plots in which character actions give rise to melodramatic conflicts. UNIVERSE was among the first to richly model explicit character qualities, e.g., intelligence and wealth. Modeling traits, emotions, and relationships has since been a key concern. A nub could encapsulate these aspects, allowing them to connect with the processes of other nubs to support coherent, ongoing narrative development. One could also help us understand how to generate single, open-ended episodes.

2.4. DAYDREAMER

Erik T. Mueller's DAYDREAMER (1987) is a cognitive model that simulates daily life experiences of its main character; these ordinary experiences trigger rich internal narratives [27]. A clear distinction between actual-world events (what happens to the character in the "real life" of a fictional world) and the character's emotional reactions to those events, such as satisfaction or disappointment when goals are achieved or blocked, is one of the key features of the system. Characters are driven by everyday human needs and goals, including self-esteem, friendship, romantic attachment, and social recognition. These motivate both real actions and imagined alternatives. In particular, the system simulates real-world rejection. Rather than focus on external plot alone, DAYDREAMER makes the character's internal cognition explicit by showing the intermediate steps of planning, evaluation, and emotional response. In its daydreaming mode, the system revisits past incidents, explores how they might have unfolded differently, and projects imagined future experiences. These alternative narratives are generated through recognizable cognitive patterns. E.g., a reversal imagines that a small change, such as wearing nicer clothes, would have led to success in asking someone on a date. Roving shifts emotional focus by recalling a different memory in which a goal was achieved. Revenge fantasies invert social roles, envisioning future success or fame. DAYDREAMER shows how creativity can emerge from fine-grained cognitive processes. A nub would have to model internal reflection, emotional reinterpretation, and imagined alternatives in response to real-world action, offering an account of how narrative meaning can arise from the interplay between lived experience and private fantasy.

2.5. GESTER

GESTER (GEnerating STories from Epic Rules, 1989) builds on Lyn Pemberton's work on the grammar of Old French epics [17]. It uses a story-grammar approach structured as a tree. Events are modeled as cause-and-effect, with explicit rules defining the relationships among characters within the story space. The two initial situations are either the lack of a wife or (interchangeably) the desire to conquer a city. As in Old French epics, a Christian man in a GESTER story can pursue any Saracen woman — married or not — while a Saracen man cannot pursue any Christian woman. We have implemented a GESTER

nub [7], with the intent of not only exemplifying the story grammar approach but also showing how characters' motivations are framed within a structured literary and cultural context.

2.6. MINSTREL

Using a story world with King Arthur and the Knights of the Round Table, Scott R. Turner's MINSTREL (1993) employs a case-based problem-solving approach and relies on two complementary types of story schemas [28]. Author-level goals guide thematic development and the creation of suspense, while character-level schemas model individual needs and motivations such as satisfying hunger or finding love. Together, these schemas represent characters, objects, emotions, the relationships between them, and the resulting story morals.

MINSTREL is the first system to explicitly model the creative writing process [5] — by storing fragments of past stories and adapting them to new narrative situations. Specifically, it employs Transform Recall Adapt Methods (TRAMs), modifying recalled story elements to avoid repetition. An example: Having a knight kill himself is a goal. Given two stored episodes, a knight killing a troll and a princess intentionally making herself ill with a potion, MINSTREL combines and adapts these so that the knight drinks a lethal potion. Some results show creativity and coherence can be difficult to maintain: Transformations can also generate nonsensical outcomes, along the lines of the "mis-spun" TALE-SPIN stories [28].

A "rational reconstruction" of MINSTREL has been done [29]; a systematic approach could allow the system's essential functions (e.g., creative transformation) to be integrated with other systems.

2.7. Grandmother

Story grammar and planning approaches are combined in Grandmother (1996) by Imogen Casebourne [30]. The generation process begins interactively, as the user is asked to define characteristics of the story, including whether the central problem should be resolved and which friendly or hostile motives (greed, revenge, etc.) should be central. Then a story grammar outlines a story, and a planner determines how characters can achieve goals. The text that results includes different-sounding sections, with *"Filled with hate for Rebekka, Emma thirsted for revenge. She resolved to hurt Rebekka"* for instance, generated by the story grammar, and *"Emma prepared to set off on her quest, but first Emma looked for Rebekka in Atlantis, however, she didn't find her there."* by planning. Given that nubs of GESTER (using a story grammar) and TAILOR (based on planning) are already implemented, Grandmother makes a good near-term project. Working to share libraries between these two systems would help reveal the extent to which the mechanisms of these approaches can be abstracted and reused. In a deeper sense, this would help us understand whether something like "story grammar" is truly better understood as a single technique or as a family of approaches with significant differences.

2.8. MEXICA

Rafael Pérez y Pérez's MEXICA system (1999) is based on an engagement–reflection cognitive account of creative story writing in which the generation of narrative content alternates with reflective evaluation in a cycle [31]. This is implemented as a production of events (in sequence) during the engagement phase and the deletion of some of them during reflection based on an assessment of novelty and interestingness. MEXICA is set in the pre-Hispanic Valley of Mexico and features indigenous characters — although story development is done using a Western creative writing model.

The system relies on knowledge structures defined as a sequence of actions, initially provided in the form of previously told stories. These establish the rules of the story world and enable the formulation of complex constraints and cultural embedding. Characters are connected through changing emotional relationships that generate tension and drive narrative progression. This tension is directly and dynamically modeled by MEXICA, aiming for rising and resolving tension arcs but also allowing situations and environments to influence characters and, in turn, be shaped by their actions. During its

evaluation phase, the system ensures that the unfolding story remains sensible, as encoded by pre- and postconditions in the knowledge structures, while also staying on an engaging tension arc.

MEXICA has been reimplemented in Java as Mexica Libre [13]. The original system has been through several iterations and there have been many spin-offs [32], one of which was MEXICA-Impro [33], in which two MEXICA agents collaborated on a story. It has also been integrated as a component within Slant [34], see Subsection 3.10. All systems of a certain maturity are actually multiple different systems from different times. MEXICA's long life makes it an ideal example of one story generator that is many. MEXICA is also interesting in how it approaches creative generation, evaluates emotional relationships, and manages tension.

2.9. BRUTUS

BRUTUS (1999) by Selmer Bringsjord and David A. Ferrucci [35] is best known for its strong focus on literary and thematic constraints rather than on surface plot mechanics or action sequencing. The system uses frame-based representations for characters, places, events, goals, and actions, using those to instantiate a theme frame, much as MINSTREL does. Then, BRUTUS develops a plot through planning and simulation, drawing on ideas from TALE-SPIN, although a grammar is used to generate the final output. BRUTUS is highly specific to the representation of betrayal. Instead of treating the plot as a chain of events driven by character goals alone, BRUTUS considers it a structure that must satisfy narrative constraints governing roles (e.g., betrayer, victim, beneficiary) and the relationships and moral tensions between them. This explicit separation between story meaning and story events is one contribution of BRUTUS. A nub could demonstrate how abstract themes and literary intentions can be encoded as constraints. In such an abstraction, it would be necessary to represent not just characters or situations, but also thematic roles and interpretive structures, allowing stories to be composed around meaning and moral tension rather than only around actions or events.

2.10. FABULIST

FABULIST (2004) by Mark O. Riedl and Robert M. Young is a planning-based plot generation system that explicitly distinguishes between the story world (with events and existents) and the discourse level, that is, how those are presented to the audience [36]. Its central goal is to harmonize author intent with character believability by integrating both author-level goals and character-level motivations into the planning process. The resulting narratives are intended to satisfy the human author's desired outcomes while ensuring that character actions remain meaningful and understandable to the reader. The system employs an Intent-driven Partial Order Causal-Link (IPOCL) planner, which combines two tightly-coupled mechanisms. The first is a partial order causal link planner that constructs chains of character actions driven by the author's global narrative goals. The second mechanism is designed to preserve character believability by simulating the audience's intention recognition process. Given a sequence of actions, the planner attempts to infer the goals a character would plausibly be pursuing. If a character's inferred goals are unclear, the actions are judged unintentional, and the plan is considered flawed, requiring revision. At the same time, FABULIST avoids making character goals too obvious, since highly predictable intentions tend to result in uninteresting stories. FABULIST demonstrates that multi-agent story planners must reason not only about how agents' actions achieve a desired story outcome, but also about how those actions signal intention to an audience. Its nub would connect the plot and discourse levels.

2.11. Curveship

Curveship (2011), a narrative discourse generator by Nick Montfort, originated as a parser-based interactive fiction system, simulating a story world with locations, characters, and objects for a player character to explore [37]. The current system omits interactive fiction capabilities and focuses on narrative variation; it does not generate plots. After specifying events that happen in the simulated world, along with parameters for narrating, the system formulates a narrative discourse. It can vary the

focalization and alter the order in which events are presented, for instance. Events are organized in trees, rather than linear sequences of events, allowing for sylleptic narration, nested flashbacks and flashforwards, and other reorganizations of events in the telling. Narration can be done from different points in time relative to the events. Such shifts in perspective and temporal ordering can change the style, tone, and interpretation of the story. From the standpoint of understanding story generators and their potential, this system is a component of one pipelined MEXICA/Curveship project and the later system Slant, see Subsection 3.10. A Curveship nub would be needed to develop a nub of those other systems; it could also be connected to nubs of other plot generators to explore automated narrating of generated plots and the effect of changes to the narrative discourse.

3. Generators on the Margins

3.1. Little Grey Rabbit Stories

A classic generator, part of a famous early computer art exhibition in 1968 [38], was the text-generating system of Eric Mendoza, used to produce essays as well as the *Little Grey Rabbit Stories*. One story begins: *"The sun shone over the woods. Across the fields softly drifted the breeze while then the clouds which calmly floated all afternoon moved across the fields."* This system is intriguing despite being called a "dead end" by a researcher who reimplemented it [39]. While not using a Markov process, it continues the text probabilistically and is purely lexical — and indifferent to whether the result is narrative. Indeed, its "stories" have little narrativity, being highly atmospheric, with animal characters performing only routine actions. Nevertheless, a nub could allow researchers to try the techniques across different genres and to combine this approach with those of other story generators, so that passages generated in this way could punctuate more eventful narratives.

3.2. Story Machine

In 1982–1983 seven different versions of an edutainment title, *Story Machine*, were released for different home computers [16]. All allowed children to type in sentences using a very limited vocabulary and grammar, then see them animated on screen. Three of them also had a WATCH A STORY mode that offered a vernacular story generator. The system allows any entity to act on any other, embodying an animisitic cosmology [40]. As with MEXICA and other research systems, it also manifested itself in several variants. The nub we implemented gave us insight into both of these aspects.

3.3. Racter

In 1984 the book *The Policeman's Beard is Half Constructed* was published [41], supposedly "[t]he first book ever written by a computer" — not the case, but a claim made by many publishers over the years [42]. Racter, a system developed by William Chamberlain and Thomas Etter, was attributed as the author of this book of poetry and prose. One story begins: "Helene embarrasses Diane during the time that she drains her champagne while a horde of expectant counsellors murder each other." A chatbot version of Racter was released for home computers. While intriguing and even positively reviewed in *The New York Times* [43], that software lacked templates and other components needed to generate the book's text. This, and the reliance on custom templates and editing, some to denounce the system (e.g., [44]). For these and other reasons, Racter has been left out of the canon. Since the 2010s, however, many have done simpler generation work than Racter did, producing short-form texts (e.g., by Twitterbots) and longer-form narrative ones (e.g., in National Novel Generation Month or NaNoGenMo) that are compelling. Racter's playful style and bizarre juxtapositions are worth study. As with the Little Grey Rabbit Stories system, Racter can work across genres and generate poetry. A book-length output and several home computer programs that can be reverse-engineered should be adequate to enable a reimplementation.

3.4. TAILOR

T. C. Smith and I. H. Witten's 1991 work on TAILOR is based on goal-directed planning by autonomous characters [18]. Inspired by game-like dynamics, their approach places characters with opposing goals into challenging situations that they must overcome by escaping or solving problems autonomously. The interaction between antagonistic anthropomorphic animals (e.g., Horace the polar bear, Truman the arctic tern) is meant to generate tension and resolution on its own. As with GESTER, a TAILOR nub has been developed, providing an example of a simple system that uses planning [7].

3.5. Repetitive but Resonant Tiny Generators

Some have developed software systems that are extremely simple, yet still might be understood as generating stories, even resonant ones. Presenting combinations of strings, they lack a story world representation and have a minimal representation of language. Creative production of this sort dates from the 1950s (when Christopher Strachey developed a love letter generator and Theo Lutz programmed *Stochastic Texts*), although these narrative projects are from much later.

Arrested: Among Nanette Wylde's Electronic Flipbooks, white-on-black textual artworks that update themselves constantly, one is clearly narrative: *Arrested,* originally from 1997, now online in JavaScript [45]. It generates sentences of the form "Some group of people were arrested for crime," for instance: *"Some Naturalists were arrested for child prostitution", "Some Lesbians were arrested for drunk driving", "Some Africans were arrested for hiding."*

Reasons for arrest need not be crimes, as in the last example. The groups of people — many of whom are marginalized and several who are named with offensive terms — are paired at random with these reasons, but a reader may sometimes rationalize or justify the arrest based on stereotypes. It's not a defect of this artwork that there is only one sort of sentence. In the story world suggested, particular groups of people are being arrested constantly.

The Two: In 2008, Montfort wrote a very short story generator in 1KB of Python, taking another Electronic Flipbook (*about so many things,* 1998 [46]) as one inspiration. His *The Two* selects one of the seven first lines at random and one of the ten last lines at random [47]. The middle line is built from a verb and two pronouns, which will, uniformly at random, be either masculine or feminine. The program outputs, for instance,

"The rescuer locates the survivalist. She surrenders to him. Each one learns something."

"The police officer nears the alleged perpetrator. He berates her. The underdog ends up on top."

To fully understand these sentences takes some thinking. It is seldom enough to understand that the subject of the first sentence is also the subject of the second. Readers have to employ their cultural and individual stereotypes about gender and power relationships.

Through the Park: Montfort released the 1KB story generator *Through the Park* at about the same time [19]. It has a sequence of 25 sentences and removes all but eight at random. A modern-day story loosely based on Little Red Riding Hood results. Gregory Weir noted in a blog comment that the inclusion or omission of "the girl's smile, the knowing glance, the blank stare, and the police siren" meant that other events "can be erotic or horrific." We recently added a nub of this system, one that supports generation in four languages.

As with many research systems, Meehan thought that to understand how TALE-SPIN works, it was important to share a small number of long stories, some about how anthropomorphic animals obtained food and one about how Joe (a human) satisfied his sexual desire. To understand these much simpler, less general systems, it is helpful to have a dozen or more examples with slight but important variations. Nubs of these systems would allow for generation and analysis of many subtly different texts which could be read in literary ways and/or compared with visual assistance.

3.6. Storyland

Wylde's 2000 *Storyland* [48] displays a changing text by assembling short paragraphs at random from three different sets of sentence building blocks. These are crafted to blend social cliches with pop-culture

references. As with the "tiny generators," the system openly displays its mechanism, in this case hinting at contemporary cultural production trends such as sampling, appropriation, and template design. *Storyland* achieves its effect by embracing absurdity through error-prone random juxtapositions and by being framed by the subtitle "Postmodern Conditions / Contemporary Tales," a circus-like title garland and musical intro.

3.7. GRIOT

GRIOT (2005) is an interactive narrative generation platform developed by D. Fox Harrell with Joseph Goguen [49]. GRIOT is concerned not with plot but poetics, meaning-making, and cultural expression across different social and cultural contexts, either purely text-based or incorporating multimedia elements such as sound and visuals. At the core of GRIOT is a computational treatment of figurative language grounded in the theory of conceptual blending [50]. Narrative meaning is constructed by integrating multiple conceptual structures into new ones, a process formalized through the Alloy algorithm. In this framework, metaphors are not explicitly authored but emerge as a side effect of blending operations over prior user-defined structured concepts. These are represented by templates containing predefined narrative structures with wildcards that can be filled dynamically based on user-input and random chance during runtime. Through this combination of template-based structure, conceptual blending, and culturally informed themes, GRIOT supports the creation of interactive multimedia narratives that center around metaphor, symbolism, and poetic interpretation rather than conventional event-driven storytelling.

A GRIOT nub would compellingly represent a very different approach to narrative generation, neither plot- or character-centered. It could contribute metaphorical framing, symbolic substitutions, and culturally grounded reinterpretations of situations generated elsewhere. Because GRIOT relies on templates with wildcards and structured themes, such a nub could accept inputs from other nubs or a user (i.e., characters, events) and re-express them through blended concepts that reflect specific cultural or poetic perspectives. The generation of many similarly themed outputs, but with different structures and contents, would also be possible, as in Harrell's "polypoems" [51]. A GRIOT-inspired system is part of Slant, see Subsection 3.10.

3.8. Dwarf Fortress Legends Mode

The cult classic *Dwarf Fortress* (2006), by brothers Zach and Tarn Adams, is an exceptionally difficult simulation and roguelike game with a visual interface that presents only ANSI characters. Each game is based on a newly generated world. As with *The Sims* (discussed in the next section), it has prompted many users to write stories based on in-game experiences. In addition to its Fortress and Adventure modes, which are more evidently game-like, there is also a Legends mode in which events in the lives of all characters, and those pertaining to artifacts, sites, and other entities are narrated. These spare event representations (e.g., *"In 77, The Submerged Quill was found in Glazedtwisted by the dwarf Thob Foughtclasped"*), can be discovered during play or offer an entry point into a newly generated world after initialization of the game. A sequence of these may seem very flat, perhaps evoking the economy of ancient chronicles. Legends data can be exported and read in an external application.

Legends mode relies on the detailed generation of a *Dwarf Fortress* world (and can reflect later play in that world). It isn't obvious how to miniaturize or abstract such a world. However, the availability of this software and ease of exporting Legends can allow for some sort of nub, potentially useful, to be developed. Nub development might help investigate how a simple narrating technique could be paired with a stripped-down (but still interesting) simulation.

3.9. Suspenser

Developed by Yun-Gyung Cheong, the 2007 Suspenser sits between the plot generation and narrative discourse generation approaches, focusing on eliciting specific cognitive and affective responses in readers, in particular suspense [52]. Rather than treating suspense as a byproduct of plot structure,

the system models it explicitly as a function of uncertainty about a protagonist's success in achieving goals and solving problems. The system takes as input a complete plan of the story's plot actions. The system first evaluates the importance of each step to form a skeletal version of the story. Suspenser then simulates the reader's interpretive process, estimating how suspenseful each potential plot can be. Given a partial plan presented in the discourse, the planner explores how many ways the protagonist can plausibly succeed. Suspense is computed as the inverse of the number of successful plans. When few solutions exist, the protagonist's success seems unlikely, heightening suspense. The system can supplement the skeletal discourse with harmful or obstructive actions that increase uncertainty.

Suspenser demonstrates how a specific narrative effect can be isolated, measured, and manipulated independently of plot generation. A nub might take an existing story plan from other nubs and work to modulate suspense. This suggests a powerful modular role for affect-driven nubs that consider the experience of readers.

3.10. Slant

The 2013 Slant by Montfort, Pérez y Pérez, Harrell, and Campana combines elements from three systems: MEXICA's engagement-reflection plot generation, narrative variation by Curveship, and figuration based on GRIOT [34]. The components are connected using a blackboard architecture, with an additional component, Verso, choosing an appropriate narrator, style, and genre for the story (i.e., diary, joke, letter) and constraining the plot if necessary. Fig-S and GRIOT-Gen (inspired by GRIOT) develop figurative expression. A specialized Curveship-Gen realizes a narrative discourse. Development presented challenges related to representational compatibility and integrating the generation process.

Slant explored how combining generators' components exposed differences and provided more expressive power. Narrative Nubs aims to facilitate similar integrations.

3.11. Scheherazade

Scheherazade (2015) by Boyang Li automates the creation of interactive text-based stories from crowd-sourced examples instead hand-authored domain models [53]. Given a topic such as a bank robbery or a romantic date, the system learns generalized plot graphs that capture relevant events of multiple linear narratives authored by human contributors. The resulting plot graph represents the legal story space to be interactively explored by the player.

Scheherazade is interesting because it shifts the burden of authoring away from manually encoding complete interactive story structures and instead uses collective human knowledge to infer them automatically. This combination of human narrative intuition with AI-based abstraction is enticing. However, there is a great deal of interactivity (from the perspective of our project) and it may not be possible to "miniaturize" a repository of human drafted stories.

3.12. Scéalextric

Scéalextric (2017) by Tony Veale develops a plot, selects main characters and their environment, and then realizes output, using an extensive knowledge-base [54]. This is based on many datasets including 2700 plot building blocks and more than 1000 real and fictional characters with 26 different attributes, creating "hybrid" stories. Given the sheer amount of data used, there is an enormous range of possibilities for stories generated by Scéalextric. The system also would help to explore the ways information about tropes and characters can be used with other sorts of plot building.

3.13. Sheldon County

While concluding his dissertation project [55], James Ryan was at work on Sheldon County, a podcast generator named after Sheldon Klein (see Subsection 2.1). The system employs several modules, including Hennepin, which models story worlds and the characters within (each with personalities, goals, and values) as they influence one another and evolve. Another module, the Expressionist, uses a

context-free grammar. The generation process includes the automated voicing of generated scripts and the generation or selection of a musical score. Importantly, a module named Sheldon is responsible not only for undertaking this work but also for rating how interesting different narratives are. Customized podcast generation was a goal; it seems that a proof-of-concept Episode 1, generated in 2018, is the one widely available output. Given this, the technically detailed description of the architecture, and the potential to connect with the author/programmer of this recently developed system, there is the potential to create a nub that embodies the several different modules and aspects of Sheldon County, each of which might be independently studied, modified, and/or integrated with other story generators.

4. ★ Distant Outlier Generators

We lack enough information about several fascinating story generators to understand them deeply as software systems. These could provide inspiration for "speculative reimplementations" — as at least one already has. Nevertheless, the scant documentation will preclude developing nubs. Other factors, including the highly interactive and sometimes graphical nature of some systems, make them "distant" from our own vantage point.

In 1960, a system called SAGA II was hastily developed at MIT by Douglas T. Ross and Harrison R. Morse for a CBS documentary "The Thinking Machine" [56]. It generated screenplays about a bank robber and a pursuing sheriff: a snippet of one printed screenplay is shown on-screen. Others were acted out, including one that was a mistaken or "mis-spun" output. The generation process was discussed in the documentary with reference to a flowchart. While an authoritative recreation of the original system would be impractical, it has already inspired a creative speculative reimplementation, SAGA III [57].

Linguist Joseph Grimes developed a system and, in 1963, used it in discussions with speakers of indigenous languages in Mexico. Contemporary documentation of this was limited to a one-page article in a trade magazine with a paragraph of output [58]. Later scholarly work revealed more [59]. We still lack information about the cast of characters, set of scenarios, and other important aspects, however.

In 1981 a three-page paper discussed an in-progress program, AUTHOR [60]. While highlighting the importance of an author's intention was helpful, little was said about how the system itself worked (or was eventually supposed to work).

A.K. Dewdney wrote a column for *Scientific American* called "Computer Recreations." In June 1989 he discussed MELL, a BASIC program of about 1500 lines by Bonnie Brunish. The column provides remarkable sample output from MELL (*"A poet bird Aeweat smells a computer Muofubumo. What is the bird coaxes Muofubumo. You are too muddled sings Aeweat. Shut up sticks the computer Muofubumo"*) and a brief description of MELL's workings. Characters were assigned descriptors that had quantitative values; interactions were based on the ones that had low values or, if such traits were lacking, the characters' occupations. A scholar contacted Brunish and, although the source code was no longer available, learned more [61], although details are still scant. A JavaScript descendant of the MELL, *Chessboard Universes*, is available on Brunish's site [62]. Given the audacious style and the way the system uniquely employs character flaws, MELL is worth including in the history of story generation. What is known about it now, however, is unlikely to support more than a speculative reimplementation.

An abstract from 2005 introduced what the authors considered the "architecture of an ideal SGA [Story Generator Algorithm]," one that modeled a distinct story level and narrative discourse as well as storytelling goals and relevant types of knowledge [63]. This could have been used as the basis for a system, but it was not a functioning software system itself.

Four systems from the 2000s would be difficult (or impossible) to abstract as nubs for different reasons. *The Sims* (2000) by Will Wright and *Maxis* and its successors generate event sequences that are represented on-screen and often resonate with users as stories; character behavior and Maslow's hierarchical needs in a user-defined environment drive these procedurally-generated narratives. This graphical and interactive game would be hard to cast as a text-based system.

Even though *Façade* (2003) by Michael Mateas and Andrew Stern [64] is surely the most frequently cited, and therefore *canonical*, interactive drama system, it poses the same problem for our purposes.

Any attempt to produce a nub of *Façade* would have to abstract away precisely those features that define it: its visual staging, voice acting, real-time dramatic interaction, and the user's active participation via free-form text input and interactions with the combative couple. Although it can certainly be regarded as a story generator, it is not central (for us) because these richly interactive and visually embodied elements, including facial expressions, are essential to the experience. Another of Mateas's systems, Subjective Avatars (1997), shows how agents can be more than neutral proxies. Here they have their own perspectives, values, emotional dispositions, and interpretations. Mateas argues that (despite not allowing pure player freedom) strong narrative experiences require meaningful constraints to increase immersion and understanding. While the system is textual, it is not an autonomous story generator, but an interactive experience that does not suit our project well.

Say Anything (2010) by Reid Swanson is also interactive; narrative is produced by textual exchange [65]. The system seeks to elicit lengthy texts from the user. While there are other essential components, a corpus of a million personal stories scraped from blogs is core. The system finds the best match to user input and continues the narrative by responding with what follows in that blog story. Some other systems have knowledge bases, but ones that are orders of magnitude smaller and can likely be abstracted and miniaturized. While we have considered Racter, Grandmother, and GRIOT, all of which involve interactivity in some ways, *The Sims, Façade,* and Say Anything require extensive user interaction throughout, making it difficult to fit them into even our own view of story generators.

5. Conclusion

Revisiting the history of story generation with an enlarged view has highlighted several issues related to narrative and the way it has been modeled, ones we hope are of general interest.

First, a model of narrative can be a mathematical formalism, a textual description in a paper (perhaps accompanied by sample output), and/or a functional software system. Our focus here was on computational models of narrative as embodied in software systems — story generators. We note that when discussing one of these with a single name (e.g., "TALE-SPIN," "MEXICA"), this name almost always actually indicates different systems that ran at different times and functioned in different ways. Reimplementing a system will always require the programmer to determine which system is really being studied or whether the reimplementation should be flexible enough to cover different versions of the same generator.

Second, as the authorized discussions of SAGA II, TALE-SPIN, and MINSTREL point out, a system's mistakes are often as interesting as its successful narrative outputs. Given that botched storytelling can create interesting literary effects, and there has been recent interest in glitch poetics, these aspects of systems deserve study through reimplementation and other means.

Next, many computer programs that generate narratives but have no clear footprint in the academic record are worthwhile. Art, literature, game, and other projects can embody bona fide models of narrative and can inform researchers' thinking about narrative.

Additionaly, a survey of diverse story generators assures us that each of them embodies a particular cosmology, a concept of how its storyworld (or universe) functions. This can resonate with our own world in compelling ways, whether or not the generator is a research system.

Finally, not every compelling story generator can be reimplemented in abstracted (or detailed) form. We knew that we would limit ourselves to rule-based/symbolic systems; among those, several are still not susceptible to our approach in Narrative Nubs. Other surveys will need to cover additional systems to provide an even richer view of story generation.

Whether or not we produce nubs of every canonical and marginal system, we hope this survey brings attention to new, worthwhile aspects of story generation systems, along with highlighting several overlooked systems that correspond to different, valuable perspectives.

Acknowledgments

Thanks to Aadya Sharma, Michelle Mo, and Roxanne Ziman for valuable comments on drafts of this paper. This research was partially funded by the Research Council of Norway Centres of Excellence project number 332643, the Center for Digital Narrative.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Gervás, Computational Approaches to Storytelling and Creativity, AI Magazine 30 (2009) 49–49. doi: 10.1609/aimag.v30i3.2250.

[2] B. Kybartas, R. Bidarra, A Survey on Story Generation Techniques for Authoring Computational Narratives, IEEE Transactions on Computational Intelligence and AI in Games 9 (2017) 239–253. doi: 10.1109/TCIAIG.2016.2546063.

[3] A. I. Alhussain, A. M. Azmi, Automatic Story Generation: A Survey of Approaches, ACM Comput. Surv. 54 (2021) 103:1–103:38. doi: 10.1145/3453156.

[4] M. Sharples, R. Pérez y Pérez, Story Machines: How Computers Have Become Creative Writers, Routledge, 2022.

[5] R. Pérez y Pérez, M. Sharples, An Introduction to Narrative Generators: How Computers Create Works of Fiction, Oxford University Press, 2023. doi: 10.1093/oso/9780198876601.001.0001.

[6] L.-Y. Bertram, N. Montfort, Output: An Anthology of Computer-Generated Text, 1953–2023, MIT Press, 2024.

[7] N. Montfort, J. Kusnick, L. Escouflaire, D. Villagran, M. Mo, A. Sharma, Narrative Nubs: Reimplementation as Inquiry into Story Generation's History, In preparation.

[8] R. C. Schank, C. K. Riesbeck, Inside Computer Understanding: Five Programs Plus Miniatures, Psychology Press, 2013.

[9] L. Carvalho, R. Martinez-Maldonado, Y.-S. Tsai, L. Markauskaite, M. Laat, How can we design for learning in an AI world?, Computers and Education: Artificial Intelligence 3 (2022) 100053. doi: 10.1016/j.caeai.2022.100053.

[10] R. C. Schank, What We Learn When We Learn by Doing. URL: https://web-archive.southampton.ac.uk/cogprints.org/637/1/LearnbyDoing_Schank.html?__hstc.

[11] P. A. Mawhorter, Artificial Intelligence as a Tool for Understanding Narrative Choices, Ph.D. thesis, University of California Santa Cruz, 2016. URL: https://api.semanticscholar.org/CorpusID:64228409.

[12] W. Sack, Spinner: A JavaScript Implementation of Tale-Spin, 2025. URL: https://github.com/warrensack/GitForNarrative, original-date: 2025-07-23T23:06:26Z.

[13] I. G. Román, R. Pérez y Pérez, A methodology to forecast some attributes of an automatic storyteller's outputs, Connect. Sci. 32 (2020) 81–111. doi: 10.1080/09540091.2019.1609418.

[14] C. Lee, prolog/talespin.pl at master · charlesjlee/prolog. URL: https://github.com/charlesjlee/prolog/blob/master/talespin.pl.

[15] L. N. Gupta, A Variant of Tale-Spin with Independent Data and Rule Bases, Master's thesis, Rochester Institute of Technology School of Computer Science and Technology, 1986.

[16] J. Schuyler, DesignWare, Story Machine, 1982. Software for six home computers.

[17] L. Pemberton, A modular approach to story generation, in: H. Somers, M. McGee Wood (Eds.), Fourth Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Manchester, England, 1989.

[18] T. C. Smith, I. H. Witten, A Planning Mechanism for Generating Story Text, Literary and Linguistic Computing 6 (1991) 119–126. doi: 10.1093/llc/6.2.119.

[19] N. Montfort, Through the Park. URL: https://nickm.com/poems/through_the_park.html.

[20] L. Polanyi, Telling The American Story: A Structural And Cultural Analysis Of Conversational Storytelling, Ablex Publishers Norwood, NJ, 1985.
[21] S. Klein, J. F. Aeschlimann, D. F. Balsiger, S. L. Converse, M. Foster, R. Lao, J. D. Oakley, J. Smith, others, Automatic Novel Writing: A Status Report, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 1973.
[22] M. Appelbaum, S. Klein, Meta-Symbolic Simulation System (MESSY) User Manual with Forward: The History of MESSY, Technical Report, University of Wisconsin–Madison, Department of Computer Sciences, 1976.
[23] J. R. Meehan, The Metanovel: Writing Stories by Computer., Yale University, 1976.
[24] W. Sack, Micro TALE-SPIN, 1992.
[25] J. Xiang, Z. Zhao, M. Zhou, M. McKenzie, A. Kilayko, J. C. Macbeth, S. Carter, K. Sieck, M. Klenk, Interleaving a Symbolic Story Generator with a Neural Network-Based Large Language Model, in: Proceedings of The Tenth Annual Conference on Advances in Cognitive Systems (ACS-22), 2022.
[26] M. Lebowitz, Creating Characters in a Story-Telling Universe, Poetics 13 (1984) 171–194. doi: 10.1016/0304-422X(84)90001-9.
[27] E. T. MUELLER, Daydreaming and Computation: A Computer Model of Everyday Creativity, Learning, and Emotions in the Human Stream of Thought, Ph.D., University of California, Los Angeles, United States – California, 1987. URL: https://www.proquest.com/dissertations-theses/daydreaming-computation-computer-model-everyday/docview/303459539/se-2?accountid=8579, publication Title: ProQuest Dissertations and Theses 8713872.
[28] S. R. Turner, MINSTREL: A Computer Model of Creativity and Storytelling, PhD Thesis, University of California, Los Angeles, 1993. URL: https://www.proquest.com/dissertations-theses/minstrel-computer-model-creativity-storytelling/docview/304049508/se-2, publication Title: ProQuest Dissertations and Theses.
[29] B. Tearse, Skald: Exploring Story Generation and Interactive Storytelling by Reconstructing Minstrel, PhD Thesis, University Of California Santa Cruz, 2018. URL: https://www.proquest.com/dissertations-theses/skald-exploring-story-generation-interactive/docview/2181579300/se-2, publication Title: ProQuest Dissertations and Theses.
[30] I. Casebourne, The Grandmother Program: A hybrid System for Automated Story Generation, in: Proceedings of the Second International Symposium of Creativity and Cognition (Loughborough, England, 1996), 1996, pp. 146–155.
[31] R. Pérez y Pérez, MEXICA : A Computer Model of Creativity in Writing, PhD Thesis, University of Sussex, 1999.
[32] R. Pérez y Pérez, MEXICA: 20 Years-20 Stories, Counterpath, 2017.
[33] R. Pérez y Pérez, S. Negrete, E. Peñalosa, R. Ávila, V. Castellanos-Cerda, C. Lemaitre, MEXICA-Impro: A Computational Model for Narrative Improvisation., in: ICCC 2010, 2010, pp. 90–99.
[34] N. Montfort, R. Pérez y Pérez, D. Harrell, A. Campana, Slant: A Blackboard System to Generate Plot, Figuration, and Narrative Discourse Aspects of Stories, in: ICCC 2013, 2013.
[35] S. Bringsjord, D. Ferrucci, Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine, Psychology Press, 1999.
[36] M. O. Riedl, R. M. Young, An Intent-Driven Planner for Multi-Agent Story Generation, in: Autonomous agents and multiagent systems, international joint conference on, volume 2, IEEE Computer Society, 2004, pp. 186–193.
[37] N. Montfort, Curveship's Automatic Narrative Style, in: Proceedings of the 6th International Conference on Foundations of Digital Games, 2011, pp. 211–218.
[38] J. Reichardt, "Cybernetic Serendipity": The First Widely-Attended International Exhibition of Computer Art : History of Information. URL: https://www.historyofinformation.com/detail.php?entryid=1089.
[39] W. Clements, Resuscitating a Dead Rabbit, in: Proceedings of EVA London 2020 (EVA 2020), 2020. doi: 10.14236/ewic/EVA2020.45.
[40] N. Montfort, A Simple 1980s Story Machine with Lessons for Narrative and Computing, Geneva, 2025.

[41] Racter, The Policeman's Beard is Half Constructed: Computer Prose and Poetry by Racter- The First Book Ever Written by a Computer, Warner Books, 1984.
[42] J. J. A. Heflin, AI-Generated Literature and the Vectorized Word, PhD Thesis, Massachusetts Institute of Technology, 2020.
[43] P. H. Lewis, PERIPHERALS; A New Brand Of Lunacy For Sale, The New York Times (1985).
[44] E. J. Aarseth, Cybertext: Perspectives on ergodic literature, JHU Press, 1997.
[45] N. Wylde, Arrested, . URL: https://preneo.org/nwylde/flipbooks/arrested/index.html.
[46] N. Wylde, about so many things, . URL: https://preneo.org/nwylde/flipbooks/aboutsomanythings/index.html.
[47] N. Montfort, The Two. URL: https://nickm.com/collab/2/the_two.html.
[48] N. Wylde, Storyland, in: Proceedings of the 29th International Conference on Computer Graphics and Interactive Techniques. Electronic Art and Animation Catalog., SIGGRAPH '02, Association for Computing Machinery, New York, NY, USA, 2002, pp. 108–109. doi: 10.1145/2931127.2931179.
[49] D. F. Harrell, Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation, in: 6th Digital Arts and Culture Conference, 2005, pp. 133–143.
[50] J. A. Goguen, D. F. Harrell, Style as a Choice of Blending Principles., in: AAAI Technical Report (7), 2004, pp. 49–56.
[51] D. F. Harrell, Walking Blues Changes Undersea: Imaginative Narrative in Interactive Poetry Generation with the GRIOT System, in: AAAI 2006 Workshop in Computational Aesthetics: Artificial Intelligence Approaches to Happiness and Beauty, 2006, pp. 61–69.
[52] Y. G. Cheong, A computational model of narrative generation for suspense (2007).
[53] B. Li, Learning Knowledge to Support Domain-Independent Narrative Intelligence, PhD Thesis, Georgia Institute of Technology, 2015.
[54] T. Veale, Déjà Vu All Over Again, in: Proceedings of the International Conference on Computational Creativity, 2017.
[55] J. Ryan, Curating Simulated Storyworlds, Ph.D. thesis, UC Santa Cruz, 2018. URL: https://escholarship.org/uc/item/1340j5h2.
[56] CBS Television Network, The Thinking Machine, 2025. URL: https://www.youtube.com/watch?v=plbPcvoHBBo.
[57] L. Daly, SAGA III—Reimplementing the first computer-generated screenplay. URL: https://lizadaly.com/pages/saga-iii/.
[58] Unknown Author, Exploring the Fascinating World of Language., Business Machines XLVI(3) (1963) 10–11.
[59] J. Ryan, Grimes' Fairy Tales: A 1960s Story Generator, in: Interactive Storytelling: 10th International Conference on Interactive Digital Storytelling, ICIDS 2017 Funchal, Madeira, Portugal, November 14–17, 2017, 2017. doi: 10.1007/978-3-319-71027-3_8.
[60] N. Dehn, Story Generation After TALE-SPIN., in: IJCAI, volume 81, 1981, pp. 16–18.
[61] Z. Whalen, A History of Chessboard Universe 50 · Issue #84 · NaNoGenMo/2020. URL: https://github.com/NaNoGenMo/2020/issues/84.
[62] B. Brunish, The Imperial Museum. URL: https://www.thermeon.net/InfernalMuseum/FIIIC2c.html.
[63] B. Lönneker, J. C. Meister, P. Gervás, F. Peinado, M. Mateas, Story Generators: Models and Approaches for the Generation of Literary Artefacts, in: ACH/ALLC 2005 Conference Abstracts. Proceedings of the 17th Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, 2005, pp. 15–18.
[64] M. Mateas, A. Stern, Façade: An Experiment in Building a Fully-Realized Interactive Drama, in: Game developers conference, volume 2, 2003, pp. 4–8.
[65] R. Swanson, Enabling open domain interactive storytelling using a data-driven case-based approach, PhD Thesis, University of Southern California, 2010.