From Text-First to Structure-First: A Visual Pipeline for Narrative Writing

From Text-First to Structure-First: A Visual Pipeline for Narrative Writing

Andrea Sillano
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Torino, Italy
andrea.sillano@polito.it

Tommaso Calò
Dipartimento di Automatica e Informatica
Politecnico Di Torino
Torino, Torino, Italy
tommaso.calo@polito.it

Luigi De Russis
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Torino, Italy
luigi.derussis@polito.it

Abstract

In narrative writing, crafting a well-structured plot requires more than ordering events, regardless of whether the work is a novel or a short story. To construct such a plot, writers typically adopt a narrative framework. However, adhering to a narrative framework can be cognitively demanding for novice writers, as traditional text-first approaches conceal the underlying structure, leaving them to manage many story elements while balancing coherence with creativity. Drawing on prior literature on narrative structuring and AI-assisted writing tools, we identified three fundamental challenges that novice writers face when constructing a plot: (1) alignment of narrative structure, (2) coherence, (3) plot ordering. We address these challenges through a visual narrative pipeline and its web-based implementation, StoryPilot, which provides a workspace to fulfill the pipeline steps by arranging narrative blocks, validating plot coherence as it evolves, and guiding alignment with the intended narrative schema.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**; *Interaction design process and methods.*

Keywords

Visual narrative planning, Writing support tool, Interactive storytelling

ACM Reference Format:
Andrea Sillano, Tommaso Calò, and Luigi De Russis. 2026. From Text-First to Structure-First: A Visual Pipeline for Narrative Writing. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3772363.3799344

1 Introduction

In narrative writing, *plot* refers to the structured organization of story events, including temporal order and causal relationship, that supports the narrative arcs [21]. Established narrative frameworks, from Aristotle's three-act framework to contemporary story grammar models, serves dual functions in plot crafting: (1) they define expectation about how stories unfold, shaping comprehension and impact [11, 20, 28], and (2) they provide cognitive scaffolds that help writers organize complex ideas into coherent plots [8, 11, 29].

Adhering to a defined narrative convention and structural form is one of the main aspects that makes crafting a coherent plot cognitively demanding, especially for inexperienced writers. Novices, indeed, must coordinate multiple interdependent elements on different cognitive layers, like character development, maintaining causal chain of events, temporal and spacial consistency, while shaping the plot to fit a defined narrative pattern [4, 34]. Narrative conventions function as an alignment between writers and audience, by establishing expectations about what matters and how events relate, thus supporting comprehension and engagement. Moreover, different narrative forms (e.g., drama, crime, TV series) rely on distinct frameworks, and writers must align plot development with the expectations of the chosen medium and genre. The tension between structural rigor and creativity often leads to a stall, where novices struggle to balance coherence and expression [4, 34]. Current workflows make these structural commitments hard to manage because planning, drafting, and revision are split across disconnected representations (e.g., notes, outlines, prompts), and errors are detected late, often after substantial text has been written.

In addition, most of the approaches fail at providing actual guidance for novice writers in constructing and revising narrative structure. According to established cognitive models of composition [3, 9], the difficulty of maintaining narrative coherence is fundamentally a representational challenge. Without an explicit representation of the narrative structure, novice writers struggle to evaluate and manipulate plot dependencies [13, 14]. In contrast, professional writers often rely on external representation like character maps or diagrams to convert abstract relationship in a more human-readable format [26, 35], lowering the cognitive load required to build a well-constructed plot [15, 18].

Together, this foundation and prior research on external representations on both writing and cognitive role in supporting planning and revision [7, 12, 18, 30] suggest that traditional text-first drafting is ill-suited for novices. In this approach, indeed, writers produce continuous prose first and only later infer or repair the plot, thus masking the narrative architecture and forcing novices to rely on an internal mental model while balancing consistency and creativity. While workflows that allows sentence-level refinement, visualization, and manipulation of plot components [10, 22, 25, 38] can reduce the cognitive load by externalizing story elements, they prioritize passive visualization and mirrors the writer's existing expertise in a graphical environment. As a result, such tools may be less supportive for novices, since they usually assume a baseline

ACM ISBN 979-8-4007-2281-3/26/04
https://doi.org/10.1145/3772363.3799344

of narrative framework competence and writing skills that they have yet to develop. To address these difficulties, we propose a visual pipeline that moves beyond visualization by making plot structure explicit and manipulable. This pipeline is implemented in StoryPilot, a web-based tool that serves as a medium to translate the pipeline steps in to a graphical user interface. The pipeline scaffolds narrative frameworks in well-defined objects with specific constrains. Writers can compose the plot by arranging and linking narrative beats, which are the core components, with each beat embedding a specific narrative function. Finally, it provides just-in-time validation of plot coherence and structural alignment as the story evolves. StoryPilot interface is composed using familiar and well-known UI interactions such as drag-n-drop, pop-up for feedback, and easy tabs navigation to fulfill the pipeline steps and to keep the focus on the narrative building task. By coupling direct structural manipulation with guidance, the StoryPilot pipeline implementation aims to reduce novices' reliance on internal mental models and help them iterate without losing narrative consistency.

2 Background

Writing a plot with a specific narrative framework require writers to follow an explicit structure (e.g., Freytag's pyramid, One-Act Play). Even when the structure is known, executing it in prose can be tedious since writers must keep track of multiple framework-specific constrains, such as what must happen when, which elements must recur, and tone consistency. From a cognitive load perspective, complex tasks can overload working memory, negatively affecting the writing process [14]. Despite the rise of AI-assisted story writing tools, most of them still treat the narrative structure as an *implicit by-product* of the generation process rather than an object that can be constructed, inspected, and corrected.

Work in this direction [2, 22, 24, 24, 27, 31] shows benefits from interacting with structural representations directly, with graph and plan-centric tools externalizing relationships or node-like workflows making high-level manipulation more concrete. These solutions support disciplined ordering by making progression an explicit design object. Other tools have different perspective, instead, relying on the ability of the writers to compose story by writing directly in prose. These systems make creation feasible by letting users assemble stories from discrete units (e.g., blocks, particles, symbols, or computational primitives) [6, 16, 17, 33]. This framing helps novices quickly construct valid artifacts and iteratively elaborate them. However, the guidance tends to be implicit in the available pieces and composition rules, so broader narrative consistency emerges from how the author connects modules over time rather than being continuously surfaced during composition. Recent LLM-enabled writing tools expand support beyond static templates by emphasizing interactive control and rapid exploration. Some primarily focus on text level co-writing [36, 37], like continuation or rewrites, while others provide light steering signals, emphasize exploring a wide space of alternatives, or support recombining fragments into emergent story directions [5, 19, 32]. Yet, the "control loop" still often runs through prompts, selection, and revision—shaping on what gets generated more than through an inspectable representation of the evolving story structure. However, cognitive models of narrative writing [9] suggest that writers do not primarily need help generating prose. Leaving the creative work to LLMs can, indeed, shift attention away from structural decision making and, as a side effect, produce more uniform plot, flattening variation and homogenizing narrative [1, 23]. Instead, novices can benefit most from support for *structural* decisions that determinate whether a sequence of scenes leads to a correct narrative structure and purposeful plot: how events are ordered, how causal dependencies remain consistent, and how the evolving draft conforms to the intended narrative both under structural and semantic perspective.

3 Narrative Design Challenges

Building on these prior works on narrative cognitive models and AI-assisted authoring we distilled three persistent challenges that repeatedly surface when writers move from ideas to plot: *alignment to narrative structure, narrative coherence, plot ordering*.

Alignment of Narrative Structure refers to a writer's ability to follow a specific narrative plot in ways that are consistent with the functional roles of structural components (e.g., exposition, rising action, climax, resolution). Common forms of misalignment include:

- *Structural omissions*: Skipping essential narrative phases entirely, resulting in deformed stories structure.
- *Functional violations*: Introducing characters or conflicts in the wrong structural phases.
- *Tonal misplacement*: Inserting humor, reflection, or relief events within structurally high-tension segments, or the opposite, introducing intense action during moments intended for setup or transition.

This challenge stems not from a lack of narrative understanding or creativity, but from the difficulty of managing structural constraints, causal relationships, and expressive intent within traditional text-based writing environments. In fact, when writing in prose, narrative structure remains implicit and fragmented across text, making it difficult for writers to know if they are correctly following the intend narrative schema. For instance a novice may open with a battle because it feels immediately "exciting," but without setup the audience has no context to grasp the stakes or invest in the outcome. Conversely, resolving the central conflict during rising action can drain the climax of its purpose, leaving the story without a meaningful peak in tension.

Narrative Coherence encompasses the challenge of maintaining consistent story elements and their relationships throughout the narrative. In traditional text-based writing, where narrative connections exist only implicitly in prose, novices frequently introduce coherence violations:

- *Broken dependencies*: Events that reference elements not yet introduced or already removed .
- *Unresolved setups*: Plot threads or character goals that disappear without resolution.
- *Disconnected beats*: Story moments that lack clear causal or thematic connection to surrounding events.
- *Entity inconsistency*: Characters or objects used in ways that contradict their established properties.

Consider a novice who introduces a mysterious artifact in the exposition but never references it again, or establishes a character's fear of water but later shows them swimming without explanation. These coherence breaks accumulate as writers lose track of the narrative elements they have introduced and their interconnections across sections.

Plot Ordering reflects the difficulty of maintaining temporal and causal relationships between narrative events. Even when novices understand global structure (exposition, rising action, climax), they struggle with local event sequencing. Common ordering violations include:

- *Inverted causality*: A character's escape described before the danger that motivated it.
- *Premature revelations*: Mystery solutions appearing before clues are established.
- *Temporal confusion*: Flashbacks that blur the story's "present" timeline.
- *Misplaced setup*: Critical world-building details introduced after dependent events.

Misplacements in structure and sequence rarely remain local. A single out-of-order event can ripple forward, forcing the writer to patch contradictions or leaving gaps that erode the story's causal logic. For example, if a character is shown using a magical artifact before the narrative establishes how it was acquired, the writer must either retrofit an explanation often awkwardly, or risk confusing readers about the world's rules. Narrative alignment, plot ordering, and coherence are tightly coupled, so difficulties in one dimension often cascade into the others. In traditional text-first writing, these relationships remain largely invisible: structure is implicit, dependencies are dispersed, and the consequences of a local edit are hard to anticipate. As a result, attempts to *fix* a story by moving an event, clarifying motivation, or adjusting pacing can inadvertently destabilize character states, causal chains, or the intended function of a section, pushing novices into repetitive revision loops. What these challenges have in common are **representational issue and lack of guidance**, rather than limitations of creativity: text alone offers limited support for simultaneously tracking structural roles, narrative dependencies, and expressive intent at the level of detail required for effective revision.

4 StoryPilot and the Visual Narrative Pipeline

To address the challenge, just described and considering the importance of the representation, we propose a visual narrative pipeline that shifts novice writing from a text-centric workflow to a structure-first, block-based drafting process with continuous validation. Our pipeline makes structural roles, causal dependencies, and evolving states visible and manipulable, allowing writers to not only visualize where possible errors originate but also enabling them to solve issues punctually inside the plot. The pipeline encodes a story as a framework-specific sequence of narrative sections. Sections are therefore constrained by the framework's intended ordering and function. Because the pipeline can use different narrative schema, the resulting storyline can vary substantially across instances. Each narrative schema is defined as a set of declarative structural constraints over an ordered sequence of sections, each characterized by a set of attributes specifying its role and function. New schemas can therefore be created and integrated by defining these constraints. Different literary frameworks define different numbers of sections, section ordering, and section functions (e.g., contextualization, inciting event, complication, turning point, consolidation, takeaway). However, once a framework is selected, the pipeline aims to preserve its internal logic: the sections are intended to appear in order and with the roles specified by that framework. Each section is associated with (i) a structural role in a target schema (e.g., exposition, rising action, climax, resolution), (ii) an evolving set of beats (e.g. characters, environments, setup), and (iii) explicit links capturing between beats. Writers compose the story by arranging, revising, and linking the beats inside the sections, while the system provides just-in-time validation that flags violations of structural alignment, coherence, or ordering as soon as they emerge. We realize a first implementation of the proposed pipeline in StoryPilot, a web application designed to show how each step of the workflow can be mapped into a set of known interaction in a graphical user interface, creating a system being able to support novices. Figure 1 reports a representation of the pipeline integrated in StoryPilot.

The visual pipeline is composed of four main steps:

Stage 1: Scaffolded Narrative. The pipeline begins with schema instantiation where writers select a target narrative schema appropriate to the task. The interface then scaffold the structure into sections tied with a specific role. Each section acts as a blank canvas where writers can develop the corresponding part of the story. Within a section, the plot is constructed through a block-based representation designed to match both the chosen schema and the section's purpose. Instead of writing continuous text from the start, writers compose the plot with reusable narrative beats. Each sections has its own sets of specific beats types designed to match that section's functional role. Writers use these beats as building units to fulfill the section's purpose (for example, exposition beats for establishing context and stakes, rising-action beats for escalation and obstacles, and climax beats for decisive confrontation). By using a literature-backed schema with role specific sections, Stage 1 makes *narrative structure alignment* explicit and reduces omissions and functional misplacements. The fixed sections sequence also establishes macro-level plot ordering before writers commit to prose. In StoryPilot, the structure scaffolding is presented as tabs inside a left side panel, where every tab shows the section's name and description and correctness status. This allows the writer to easily navigate through the sections' schema and also to become aware of errors' position inside the composition.

Stage 2: Structured Beat Authoring. After instantiating the narrative framework, the writers can start filling the section's canvas. Narrative beats serve as the core units writers can use them to construct a story. Each beat encodes a specific narrative function, such as introducing a character or establishing the setting. Instead of directly drafting prose immediately, novices first externalize the story's structure by assembling these modular elements into a coherent progression and filling the beats information, thus creating an explicit scaffold that can be inspected and revised before committing to full text. Writers can add, edit, duplicate, and most importantly can connect narrative beats encoding relationships that impose structural constraints. By linking beats, writers make dependencies explicit. These connections preserve the intended flow of information and causality, constrain invalid re-orderings, and allow for early detection of missing prerequisites. Stage 2 supports *plot ordering and narrative coherence* by letting writers assemble and connect beats through visual dependencies. Sections specific beat further reinforce alignment by steering writers toward

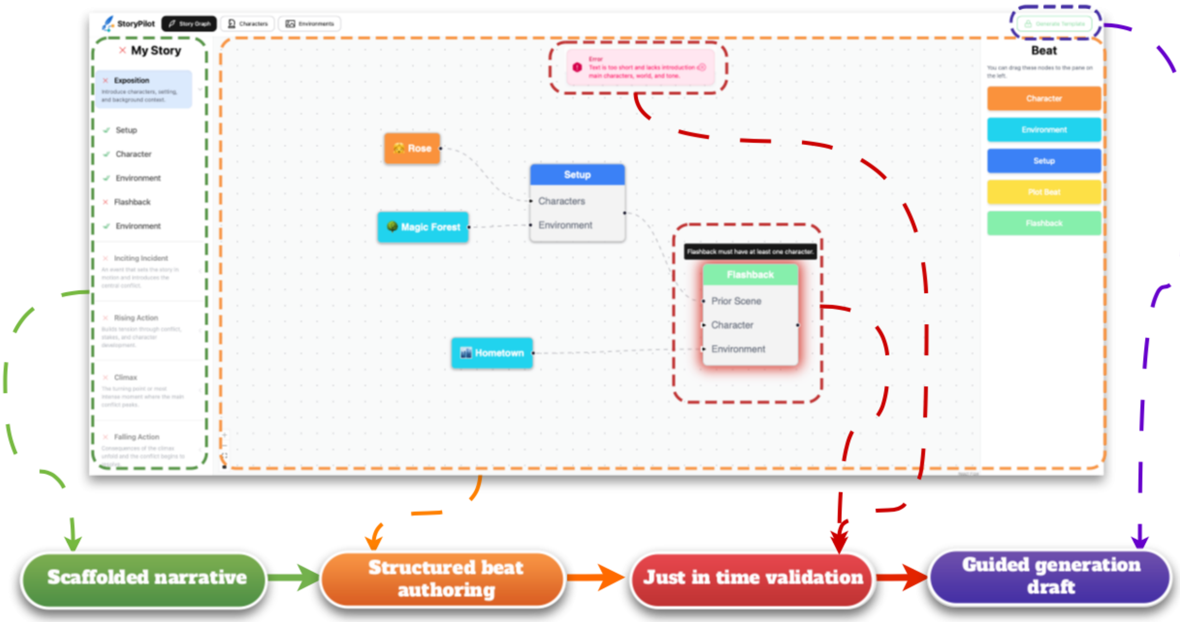

Figure 1: A visual representation of the implementation of the pipeline inside StoryPilot.

structurally appropriate narrative moves. StoryPilot supports beat authoring through a dedicated section canvas that serves as the primary workspace for plot development. Within this canvas, writers can drag and drop narrative beats to assemble a section's progression and by filling the beats define the section's content, they con quickly sketch alternative arrangements, and refine the flow as the storyline takes shape. Each beat acts as a structured building block, helping writers move from high-level planning to concrete plot composition without losing track of the section's intent.

Stage3: Just in time Validation. The core novelty of this pipeline is the just-in-time validation feedback mechanism delivered while writers are manipulating the structure and the contents. Validation operates on three different levels: sections, beats content and beats connection. Section and connection checks can be enforced through declarative structural constraints on the beat graph, while beat-content checks can use LLMs to provide semantic feedback and revision guidance.

- *Section-level validation*: checks each section against the target schema to identify missing required beats and purpose misalignment. It validates if the content, written by the author, in the beats conflicts with the section's functional role.
- *Beat-level content*: verifies whether the beat's configured content fulfills correctly the beat's intended purpose.
- *Beat-level links*: evaluates local links between beats to ensure they respect both narrative constraints (coming from narrative framework) and causal consistency.

Stage 3 addresses the three challenges through layered feedback: "section checks" flag alignment gaps and role conflicts, "beat checks" ensure each beat's content, fulfills its intended function, while "connection checks" enforce causal ordering and narrative constraints across linked beats. StoryPilot implements the just in time validation mechanism through different methods based on levels. At section level, the application represent the status as a red cross or a green check-mark icon near the section name, drawing the attention of the writer to quickly visualize where the issues are located. The beat's content and linking share the same goal of providing feedback, in particular beats with errors are contoured with a red or yellow halo based on the gravity. While the content feedback is shown as a red popup on top of the screen containing the errors description and guidance on how to possible solve them, since coherence and misalignment can be found in multiple beats and require a more in-depth understanding. The linking issues are presented as a tool-tip on top of the beats showing what's the cause, enabling writers to quickly identify the solution. **Stage 4: Guided Generate Draft**. Finally, the pipeline supports translating the structured beat representation into a guided draft while preserving the connection between text and structure. Rather than producing a finished story, the output is a prose draft augmented with writing guidance: each section is realized as a scaffolded passage that conveys what should be written next, what information must be included, and how the content should function within the target schema. The beat structure acts as the backbone of the drafting and even after the text is generated, writers can still reorder beats, adjust sections or expand the story. Stage 4 help to ease the burden of planning, editing and reviewing so writers can focus on creative choices while still staying aligned with a narrative framework. In StoryPilot, once the writer finishes composing the plot and all issues are resolved, the generate button becomes available, allowing the system to produce the draft from the current plot e.g., via LLM-based generation conditioned on the structured beats. The writer can visualize the composition and if not satisfied can go back to the

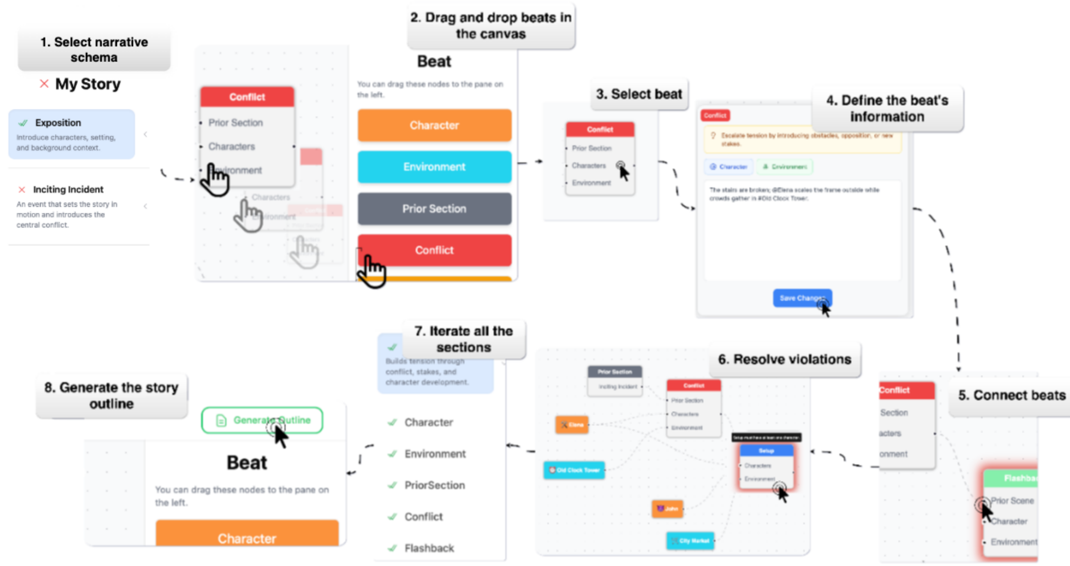

Figure 2: A step-by-step walkthrough of the pipeline's stages inside StoryPilot.

canvases and edit it, otherwise can download the draft containing all the guidance and information.

Figure 2 shows StoryPilot's walkthrough of the pipeline's stages. Writers first select a narrative schema, which defines the sections and their intended functions. They then can drag-and-drop story beats onto a visual canvas to shape the plot. For each beat, writers add key details and connect beats to encode causal relationships. Throughout this process, StoryPilot provides just-in-time structural and semantic feedback, flagging issues such as missing links or inconsistent logic so writers can revise in place. Writers then progress section-by-section according to the chosen schema until all sections pass validation, after which the system compiles the validated structure into a scaffolded outline-guided draft ready to be expanded into full prose.

5 Conclusion and future work

In this work, we introduced a structure-first visual narrative pipeline and StoryPilot, a web-based tool that scaffold plot development around a selected narrative schema. The pipeline encodes plots as a specific sequences of sections, and provides in-section canvases where writers assemble and connect narrative beats, making structural roles and causal dependencies explicit and editable. It also introduce a three-level feedback system and allows for the generation of plot draft aligned with a target framework. By externalizing the structure and giving guidance during plot composition, this pipeline aims to reduce the cognitive effort needed and to manage the writing process. The visual representation offered by StoryPilot can help novices to offload the planning, translating, reviewing tasks by working directly on the interface rather than on prose. Collectively, the visual narrative pipeline and StoryPilot aim to support novices in plot development, and also point to broader implications for the design of future writing support tools.

References

[1] Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization Effects of Large Language Models on Human Creative Ideation. In *Creativity and Cognition (C&C '24)*. ACM, 413–425. doi:10.1145/3635636.3656204

[2] Victor Nikhil Antony and Chien-Ming Huang. 2024. ID.8: Co-Creating Visual Stories with Generative AI. *ACM Transactions on Interactactive Intelligent Systems* 14, 3, Article 20 (Aug. 2024), 29 pages. doi:10.1145/3672277

[3] Carl Bereiter and Marlene Scardamalia. 1987. *The Psychology of Written Composition.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[4] Paul Chandler and John Sweller. 1991. Cognitive Load Theory and the Format of Instruction. *Cognition and Instruction* 8, 4 (1991), 293–332. doi:10.1207/s1532690xci0804_2

[5] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching Stories with Generative Pretrained Language Models. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 19 pages. doi:10.1145/3491102.3501819

[6] John Joon Young Chung, Melissa Roemmele, and Max Kreminski. 2024. Toyteller: Toy-Playing with Character Symbols for AI-Powered Visual Storytelling. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '24)*. Association for Computing Machinery, New York, NY, USA, 5 pages. doi:10.1145/3672539.3686781

[7] Stephen Ciullo and Colleen Reutebuch. 2013. Computer–Based Graphic Organizers for Students with LD: A Systematic Review of Literature. *Learning Disabilities Research & Practice* 28, 4 (2013), 196–210. arXiv:https://journals.sagepub.com/doi/pdf/10.1111/ldrp.12017 doi:10.1111/ldrp.12017

[8] Jill Fitzgerald and Alan B. Teasley. 1986. Effects of instruction in narrative structure on children's writing. *Journal of Educational Psychology* 78, 6 (1986), 424–432. doi:10.1037/0022-0663.78.6.424

[9] Linda Flower and John R. Hayes. 1981. A cognitive process theory of writing. *College Composition and Communication* 32, 4 (1981), 365–387. doi:10.2307/356600

[10] Parsa Ghaffari and Chris Hokamp. 2025. Narrative Studio: Visual narrative exploration using LLMs and Monte Carlo Tree Search. arXiv:2504.02426 [cs.AI] https://arxiv.org/abs/2504.02426

[11] C. G. Glenn. 1980. Relationship between story content and structure. *Journal of Educational Psychology* 72, 4 (1980), 550–560. doi:10.1037/0022-0663.72.4.550

[12] Matthias Grünke, Marko Sperling, and Mack D. Burke. 2017. The Impact of Explicit Timing, Immediate Feedback, and Positive Reinforcement on the Writing Outcomes of Academically and Behaviorally Struggling Fifth-Grade Students. *Insights into Learning Disabilities* 14, 2 (2017), 135–153. https://eric.ed.gov/?id=EJ1164944 ERIC Number: EJ1164944.

[13] R.T. Kellogg. 2008. Training writing skills: A cognitive developmental perspective. *Journal of Writing Research* 1 (06 2008), 1–26. doi:10.17239/jowr-2008.01.01.1

[14] Ronald T. Kellogg. 2001. Competition for Working Memory among Writing Processes. *The American Journal of Psychology* 114, 2 (2001), 175–191. http://www.jstor.org/stable/1423513

[15] David Kirsh. 2010. Thinking with external representations. *AI & Society* 25, 4 (2010), 441–454. doi:10.1007/s00146-010-0272-8

[16] Varsha Koushik, Darren Guinness, and Shaun K. Kane. 2019. StoryBlocks: A Tangible Programming Game To Create Accessible Audio Stories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 1–12. doi:10.1145/3290605.3300722

[17] Xingyu Lan, Yiran Shi, Yanhong Wu, Xiaohan Jiao, and Nan Cao. 2023. DataParticles: Block-based and Language-oriented Authoring of Animated Unit Visualizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, 1–18. doi:10.1145/3544548.3581472

[18] Jill H. Larkin and Herbert A. Simon. 1987. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science* 11, 1 (1987), 65–100. doi:10.1111/j.1551-6708.1987.tb00863.x

[19] Zhuoran Lu, Qian Zhou, and Yi Wang. 2025. WhatELSE: Shaping Narrative Spaces at Configurable Level of Abstraction for AI-bridged Interactive Storytelling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 333, 18 pages. doi:10.1145/3706598.3713363

[20] J. M. Mandler and N. S. Johnson. 1977. Remembrance of things parsed: Story structure and recall. *Cognitive Psychology* 9, 1 (1977), 111–151. doi:10.1016/0010-0285(77)90006-8

[21] Inderjeet Mani. 2013. *Plot.* Springer International Publishing, Cham, 73–94. doi:10.1007/978-3-031-02147-3_4

[22] Damien Masson, Zixin Zhao, and Fanny Chevalier. 2024. Visual Writing: Writing by Manipulating Visual Representations of Stories. *arXiv preprint arXiv:2410.07486* (2024). doi:10.48550/arXiv.2410.07486

[23] Kibum Moon, Adam E. Green, and Kostadin Kushlev. 2025. Homogenizing effect of large language models (LLMs) on creative diversity: An empirical comparison of human and ChatGPT writing. *Computers in Human Behavior: Artificial Humans* 6 (2025), 100207. doi:10.1016/j.chbah.2025.100207

[24] Eric Mörth, Stefan Bruckner, and Noeska N. Smit. 2023. ScrollyVis: Interactive Visual Authoring of Guided Dynamic Narratives for Scientific Scrollytelling. *IEEE Transactions on Visualization and Computer Graphics* 29, 12 (2023), 5605–5617. doi:10.1109/TVCG.2022.3205769

[25] Jeongyoon Park, Jumin Shin, Gayeon Kim, and Byung-Chull Bae. 2023. Designing a Language Model-Based Authoring Tool Prototype for Interactive Storytelling. In *Interactive Storytelling*, Lissa Holloway-Attaway and John T. Murray (Eds.). Springer Nature Switzerland, Cham, 239–245.

[26] Phyllis Taylor Piper. 2009. *The Fiction Writer's Toolkit: A Guide to Writing Novels and Getting Published.* Aspen Mountain Press, Denver, CO.

[27] Hua Xuan Qin, Guangzhi Zhu, Mingming Fan, and Pan Hui. 2025. Toward Personalizable AI Node Graph Creative Writing Support: Insights on Preferences for Generative AI Features and Information Presentation Across Story Writing Processes *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 897, 30 pages. doi:10.1145/3706598.3713569

[28] Daniel S. L. Roberts, Paul S. Cowen, and Brenda E. Macdonald. 1996. Effects of Narrative Structure and Emotional Content on Cognitive and Evaluative Responses to Film and Text. *Empirical Studies of the Arts* 14, 1 (1996), 33–47. arXiv:https://doi.org/10.2190/1L6D-FA7K-UQ0V-B7UM doi:10.2190/1L6D-FA7K-UQ0V-B7UM

[29] David E. Rumelhart. 1975. NOTES ON A SCHEMA FOR STORIES. In *Representation and Understanding*, DANIEL G. BOBROW and ALLAN COLLINS (Eds.). Morgan Kaufmann, San Diego, 211–236. doi:10.1016/B978-0-12-108550-6.50013-6

[30] Mike Sharples and Lyn Pemberton. 1992. *Representing Writing: External Representations and the Writing Process.* Springer Netherlands, Dordrecht, 319–336. doi:10.1007/978-94-011-2854-4_21

[31] Yotam Shibolet and Hartmut Koenitz. 2022. Storygraphia: The Constrained Tool for IDN Authoring Education. In *Interactive Storytelling: 15th International Conference on Interactive Digital Storytelling, ICIDS 2022.* Springer, 591–596. doi:10.1007/978-3-031-22298-6_38

[32] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. ACM, 1–26. doi:10.1145/3613904.3642400

[33] Sangho Suh, Jian Zhao, and Edith Law. 2022. CodeToon: Story Ideation, Auto Comic Generation, and Structure Mapping for Code-Driven Storytelling. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 13, 16 pages. doi:10.1145/3526113.3545617

[34] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science* 12, 2 (1988), 257–285. doi:10.1207/s15516709cog1202_4

[35] John Truby. 2007. *The Anatomy of Story: 22 Steps to Becoming a Master Storyteller.* Faber and Faber, New York, NY.

[36] Zhen Wu, Serkan Kumyol, Shing Yin Wong, Xiaozhu Hu, Xin Tong, and Tristan Braud. 2025. Orchid: A Creative Approach for Authoring LLM-Driven Interactive Narratives. In *Proceedings of the 2025 Conference on Creativity and Cognition*. Association for Computing Machinery, New York, NY, USA, 774–791. doi:10.1145/3698061.3726906

[37] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story Writing With Large Language Models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 841–852. doi:10.1145/3490099.3511105

[38] Zixin Zhao, Damien Masson, Young-Ho Kim, Gerald Penn, and Fanny Chevalier. 2025. Making the Write Connections: Linking Writing Support Tools with Writer's Needs. In *CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 23 pages. doi:10.1145/3706598.3713161