Working with Intelligent Narrative Technologies

David Thue

Abstract Artificial Intelligence systems have been used to generate narrative structures and simulate virtual story characters at a variety of different scales, across both academia and industry. Such systems are often built from specialized components known as intelligent narrative technologies. The goal of this chapter is to highlight some of the challenges that can arise when such technologies are used as part of authoring or executing an interactive story. Authoring in a way that works with these technologies often requires a host of technical skills, such as writing computer code, building mathematical models, or predicting the effect of a simple change on a large, complex system. In addition to explaining why these skills are needed and the problems that they help to solve, this chapter will highlight recent and ongoing efforts to make authoring for intelligent narrative technologies more accessible to those with fewer technical skills.

1 Intelligent Narrative Technologies

The phrase "intelligent narrative technologies" can have (at least) three meanings. One is that it describes a field of research, which studies how the techniques used by Artificial Intelligence (AI) systems can be applied in the context of narrative. While there are examples of such research from the early 1960s [1], it became more widespread in the 1990s [2–6] and continues actively to this day. Mateas and Sengers offer a detailed account of the early years of this research field in the first chapter of their book [7].

The second meaning of "intelligent narrative technologies" is that it is the name of a series of academic events, which began in 2007 [8] and was held most recently in 2020 [9]. In total, these events included two research symposia [8, 10], several workshops co-located with three academic research conferences [9, 11–18], and a special track at the International Conference on Interactive Digital Storytelling [19].

D. Thue (✉)
RISE Research Group, School of Information Technology, Carleton University, Ottawa, Canada
e-mail: david.thue@carleton.ca

Department of Computer Science, Reykjavik University, Reykjavik, Iceland

C. Hargood et al. (eds.), *The Authoring Problem*, Human–Computer Interaction Series,
https://doi.org/10.1007/978-3-031-05214-9_17

For the past 15 years, these events have been a common home for early-stage research done in the field of intelligent narrative technologies. A related series of events featured a recurring workshop on Computational Models of Narrative [20–26].

The third meaning of the phrase is more pragmatic, and it is the one that we focus on in this chapter: *intelligent narrative technologies* (INTs) are technologies that apply AI techniques in the context of narrative. They are the focus and products of the research done in the field of INT, and the primary topic of the papers that are published via the INT series of events.

What does it mean, then, to apply AI techniques in the context of narrative? Fundamentally, AI techniques can be applied to make decisions in an automated way, and working in narrative means making decisions in that context. In Interactive Digital Narrative (IDN), many decisions have been made using AI techniques, and these decisions have centered primarily on the potential *products* of an IDN system [27]. They answer questions that include (but are not limited to):

- *What characters and objects should exist in the narrative world?*
- *What should happen next in the story?*
- *What should this character do next?*
- *How should this character perform its next action or line of dialogue?*
- *How should the system respond to the player's last action?*

The methods that have been used to answer these questions are many and varied, and citations to works that explain some of them will appear throughout this chapter. The focus of this chapter, however, is different: rather than explain how a collection of INTs work, it aims to equip IDN authors with general strategies that might help them work more effectively with intelligent narrative technologies.

1.1 Authoring with a Narrative AI System

For the purposes of this chapter, we consider *authoring* to be a process of making and acting upon decisions about how some elements of a narrative (or perhaps many possible narratives) should be. This could involve creating characters, locales, key props, storyboards, and more. Furthermore, we consider a *narrative AI system* as a structured collection of one or more intelligent narrative technologies, each of which might apply different AI techniques; the system accepts one or more inputs (some provided by authors in advance, others provided by players at run-time) and produces one or more outputs using the technology therein. For example, the AI-driven "drama manager" in *Façade* [28] accepts inputs including (i) a collection of dramatic beats (bundles of narrative content), (ii) an estimate of the story's current level of dramatic tension, and (iii) an authored trajectory of dramatic tension over time [29]. Given these inputs, it uses an optimization technique to identify a particular dramatic beat from the collection as its output: the one that best matches the next author-desired level of tension in the story.

It is common to say that an AI system "decides" which output(s) it should produce as it operates. Since authoring is about making and acting upon decisions, we say that an authoring process can *include* a narrative AI system; in such cases, the decisions that are made during authoring will be shared between the author(s) and the system. From an author's perspective, it can thus be useful to understand what drives a narrative AI system's decisions, along with how those decisions can be influenced. To build such an understanding, an author can pursue answers to the following key questions. We discuss strategies for tackling these questions as the core content of this chapter.

- How does the AI system behave?
- How can I influence the AI system's behaviour?
  - How can I determine the AI system's inputs?
  - What of the AI system itself can I change?
  - How can I refine or repurpose the AI system's outputs?

2 Understanding the Behaviour of a Narrative AI System

What can an author do to understand how a narrative AI system behaves? Following some suggestions for initial preparation, we discuss two types of strategy: experimentation and examination.

2.1 Preparation: Understand the IDN System

Fundamentally, every IDN system requires a *protostory* [27], which represents what exists in the narrative world, the properties of those objects, and how they can change during a player's experience—either in response to player input or due to the passage of time. Narrative AI systems act upon this protostory, either by helping to define its elements before any player's experience begins or by steering how the narrative world changes as each experience unfolds. When attempting to understand a narrative AI system, it can therefore help to first learn about the IDN system's protostory. Given this knowledge, the author can approach the AI system by first assessing which parts of the protostory the AI system is used to determine or change, and then applying the strategies in Sects. 2.2 and 2.3 to learn about each part. An example that explains *Façade*'s protostory and how *Façade*'s drama manager changes it can be found in prior work [27], and we will revisit it before the end of this section.

In addition to understanding how a narrative AI system might affect an IDN system's protostory, it can be helpful to know how a player's actions might affect the AI

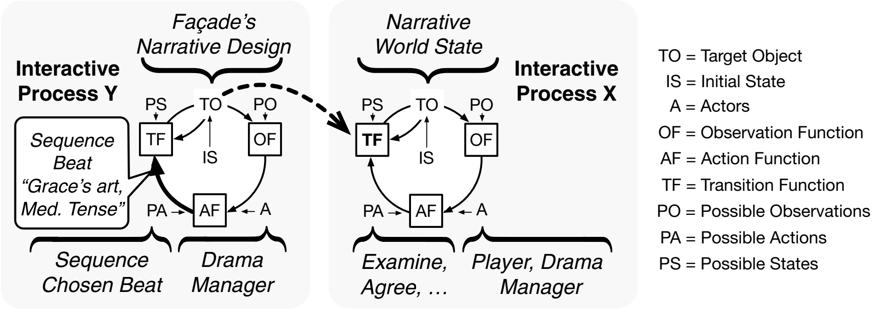

Fig. 1 A partial interactive process model [30] for *Façade* [28], where both the player and *Façade*'s drama manager are modelled as agents. Each interactive process models how a particular Target Object (TO) can be changed by the Actors (A) that participate in that process. Actors execute the Action Functions (AF) to perform specific actions, and the IDN system executes the Transition Functions (TF) and the Observation Functions (OF) to produce, respectively, new states of each Target Object and new specific observations for each actor to observe. For a complete presentation of Interactive Process Modelling, see Chap. 'Getting Creative with Actions'

system. A method for understanding player actions in IDN systems (Interactive Process Modelling) is presented in Chap. 'Getting Creative with Actions'. By building an interactive process model for the IDN system that treats both players and narrative AI systems as agents, an author can visualize and reason about how a player's actions might influence the behaviour of a narrative AI system. Figure 1 shows an example of such a visualization for *Façade*.

The player performs actions via the action function (AF) in Interactive Process (IP) X, such as examining an object or agreeing or disagreeing with characters Grace or Trip. As they do so, the state of the narrative world changes and the result can be observed by both themselves and the drama manager (via the observation function, OF, in IP X). Meanwhile in IP Y, the drama manager is able to use what it previously observed in IP X to inform how it chooses which dramatic beat should occur next. By executing IP Y's action function (AF) to make this choice, it alters *Façade*'s narrative design, which is the part of the protostory that represents how the world changes during player experiences. *Façade*'s narrative design is both the target object (TO) of IP Y and the transition function (TF) of IP X, as shown by the figure's *target object link* (dashed arrow). This link models how the manager's action to sequence a subsequent dramatic beat (which happens in IP Y) ultimately affects what beat occurs next in IP X, as the player's experience in IP X proceeds.

By understanding a player's ability to influence a narrative AI system, authors can be better equipped to anticipate the effects that player interaction might have on their uses of that system.

2.2 *Experimenting with a Narrative AI System*

The first method for system-learning that we consider is experimentation. This method of learning requires having access to the narrative AI system in a way that lets the author trigger and observe the results of system decisions under a variety of different circumstances. This might involve using support tools created by the system's developers. For example, *Sentient Sketchbook* allowed authors to see examples of the maps that it could generate and tune parameters that affected how they were generated [31], while *Mimisbrunnur* allowed authors to view potential sequences of narrative action that could occur during gameplay, given the content and constraints that the author had created [32]. Experimentation might also involve playtesting an IDN process that the narrative AI system influences, to bring the narrative world state into particular situations and observe how the AI system reacts. For example, much can be learned about *Façade*'s drama manager [33] by playing *Façade* and observing how the AI system behaves [28].

In general, an author can produce different circumstances for a narrative AI system by modifying the system's inputs. Depending on what INTs the system uses, these inputs might include collections of various kinds of content, or parameters or utilities. We consider some examples of each.

2.2.1 Collections of Content as System Inputs

It is quite common for a narrative AI system to require one or more collections of content among its inputs, and the types of required content can vary widely across different systems.

As one example, several narrative AI systems that produce natural language text (*e.g.*, for character dialogue) require large collections of text to be provided as inputs; such collections might include film scripts, blog posts, news articles, and more. At a high level, such systems contain one or more INTs that perform natural language generation—they use the provided collections of text to build a general model of how people tend to write sentences and paragraphs in a given context (*e.g.*, in Science Fiction movie scripts), and then use the model to predict the words of new sentences and paragraphs. A compelling IDN system that uses this sort of technology is *AI Dungeon* [34].

Collections of images or 3D models are common inputs to narrative AI systems—particularly for those that are embedded in video games and generate some of their game's content. Procedural Content Generation (PCG) describes a process of automatically creating content (typically to be used in a game). When applied to narrative contexts, PCG methods become INTs, as they are used to make decisions about how narratively-important content should be. Examples include the generation of non-player characters (including their appearance and attributes) [35], towns [36], and more. Two compelling IDN systems that use PCG to create narrative content are *Dwarf Fortress* [37] and *RimWorld* [38].

A frequently studied subtopic of INT research is Narrative Planning [39], which uses automatic, logical reasoning to find plans of action for story characters that satisfy goals given by an author. To form such plans, the narrative planner (which is a narrative AI system) requires a collection of potential actions for characters to perform, plus a collection of characters and other entities (*e.g.*, props and locations) whose attributes can be changed by the given actions. For example, the authoring tool *Mimisbrunnur* allowed authors to create collections of actions and entities, and then preview examples of how a narrative planner might use those actions to achieve different story goals [32].

By adjusting the collections that a narrative AI system receives in its inputs, an author can put the system in different circumstances and observe how it behaves therein.

2.2.2 Parameters and Utilities as System Inputs

Many narrative AI systems have *parameters*—variables that are meant to be adjusted (sometimes by players, sometimes by designers) to alter its behaviour. For example, at the beginning of a game of *RimWorld*, players are able to set a variety of parameters that control how the game's narrative AI system will behave [38]. The system generates notable events from a library of templates (*e.g.*, attacks by hostile creatures or extreme weather), and the parameters affect various aspects of how those events get generated (*e.g.*, their frequency or severity).

One weakness of parameters is that they each remain fixed at their given value, regardless of what might happen while the system is operating. When it is important for a variable's value to change in response to changing circumstances, a narrative AI system might require an input that helps it compute new values for that variable. This sort of input can be well thought of as a *utility*—something that allows new values of a variable to be calculated given other values (*e.g.*, of other variables, or of attributes of entities in the narrative world). For example, the drama manager in *Façade* [28] required a designer-provided utility that does the following: It starts by considering the history of the story's events thus far, and retrieves estimates of how each event should have contributed to the story's dramatic tension. Then, by adding these contributions together, the utility produces an estimate of the *current* state of dramatic tension in the story. This utility is used regularly during gameplay to estimate the story's current tension level [29].

In *RimWorld*, players can choose between three "AI storytellers" [38], each of which activates a separate collection of parameter values and utilities that were specified by the game's designers. In particular, each collection has a utility to measure the player's level of success (*e.g.*, based on their accumulated wealth in-game) as well as a utility to determine how soon the next dangerous event should occur, given the player's current level of success.

2.2.3 Understanding versus Complexity

The goal of experimenting with a narrative AI system is for the author to build a useful understanding of how that system behaves. The degree to which such experimentation is effective, however, depends on the complexity of the INTs that are used in the AI system and the amount of precision the author needs. For example, the INTs used by *AI Dungeon* [34] are *transformer neural networks*, which are models of English language usage that come from an enormous corpus of written text [40]. These models are capable of generating coherent English prose, but their internal workings are so complex (with millions to billions of parameters) that understanding them perfectly is all but impossible. Nevertheless, a large community of users has gained a sufficient understanding of *AI Dungeon* to author their own narrative scenarios [41]. Given the generative nature of *AI Dungeon* experiences, even an imprecise understanding of its models can allow authors to set up interesting player experiences.

2.3 Examining a Narrative AI System

Beyond observing a narrative AI system to gain an impression of how it behaves, authors who are more technically inclined may be able to examine the system itself, toward learning how its internal mechanisms lead it to behave in different ways. Such an examination might involve reading publications or other technical documents, or reading the program code that executes while the system operates.

When a narrative AI system is described across multiple publications of different types and lengths, it can be challenging to know where to start. Research papers published at academic workshops or conferences, as well as white papers and blog posts written by developers, typically contain high level explanations in a relatively compact presentation. These works can be useful for identifying the required inputs of a given system, but many also contain a concise description of how the system works, either through figures, pseudocode, or both. Writing in pseudocode allows a system developer to express the key steps of the system's program code in a way that is more readable than the program code itself, and pseudocode should ideally be understandable even with only a limited knowledge of computer programming.

As an example, Algorithm 1 shows pseudocode that explains the operation of a kind of narrative AI system called a *player-specific experience manager*[1] [43]. The inputs and expected output are stated at the top. On Line 1, a current model of the player (*e.g.*, representing their preferences) is updated based on the current narrative world state and the player's most recent action. On Lines 2 to 7, the set of possible adaptations is searched, element by element, while estimating how well each possible adaptation matches with the current player model (Line 4). Each time a better match is found, it is set as the next adaptation that should be performed to adapt the current

[1] The term *experience manager* is due to Riedl et al. [42], and refers to an AI system that attempts to modify the course of a player's experience as it proceeds.

Algorithm 1: Pseudocode showing the high-level operation of a narrative AI system that adapts a story using a learned player model. Italics show variables and upright text shows utilities that compute useful values. The notation '$x \leftarrow y$' means 'x gets set to the value of y'.

```
Inputs  : narrativeWorldState: the current state of the narrative world
          playerAction: the most recent action that the player performed
          currentPlayerModel: prior information learned about the player
          possibleAdaptations: a collection of ways to adapt the current story
Outputs: nextAdaptation: the adaptation that best matches with the player model

1 currentPlayerModel ←
  GetUpdatedPlayerModel(currentPlayerModel, narrativeWorldState, playerAction)
2 bestMatchQuality ← 0
3 for each adaptation in possibleAdaptations do
4     matchQuality ← EstimateMatchQuality(adaptation, currentPlayerModel)
5     if matchQuality > bestMatchQuality then
6         bestMatchQuality ← matchQuality
7         nextAdaptation ← adaptation
8
9 return nextAdaptation
```

story (Line 7). The result is that the adaptation that matches the player model the best is the one that will be output. Although this example has been simplified for the sake of introducing the concept (notably by avoiding most mathematical notation), it nonetheless demonstrates the rough character of how pseudocode is presented in technical writing about narrative AI systems.

For authors who have the needed technical background to understand program code directly, some narrative AI systems can be understood in depth by requesting the code from its authors or by finding it in a public repository online. A recent example is *Imaginarium* [44], a casual tool for generating narrative content (*e.g.*, characters or objects) whose source code was made available online [45].

Given the ability to read its pseudocode or program code, an author can directly examine the operation of a narrative AI system, and use what they discover to understand the system's general behaviour.

3 Ways to Influence a Narrative AI System

Once an author has come to understand how a narrative AI system might behave, their attention might turn to the question of how they can make it behave *differently*. In this section, we consider the remainder of the questions that we asked in Sect. 1.1, including how an author might determine the inputs of a narrative AI system, how they might alter the system itself, and how they might refine or repurpose what the system produces as output.

3.1 Determining the System's Inputs

As we discussed in Sect. 2.2, it is often possible for an author to change various inputs of a narrative AI system. Indeed, the majority of the inputs of many narrative AI systems are expected to be authored by one or more people. For example, *RimWorld*'s world generator (another narrative AI system) can generate an entire unique planet, including terrain, biomes, creatures, settlements, and inhabitants. From a player's perspective, it seems to generate all of this from very little: a single starting seed (a random string of characters) and a handful of generation parameters. In reality, the world generator works by cleverly combining many collections of pre-authored content, including trees, boulders, creatures, building materials, props, character attributes, character appearances, and more. The starting seed and generation parameters matter, but the pre-authored content provides the bulk of the resources that are used while the generator works. As a result, one way to influence how *RimWorld*'s world generator behaves could be to modify one or more of these collections of content, all of which the generator uses as inputs. Other ways include changing either the starting seed or the generation parameters.

An important difference can be seen between these methods of exerting influence: some are more *controllable* than others, in that the outcome of any change is easier to predict and thus easier to use in intentional ways. While the effects of replacing a boulder in the pre-authored content seems relatively easy to predict and thus more straightforward to control, the effects of changing a starting seed are nearly impossible to predict. This makes the seed value more challenging to use when pursuing particular authorial aims.

Beyond parameters and collections of content, it is also often expected that the utilities that are required by a narrative AI system will be authored by one or more people. For example, the trajectory of dramatic tension over time that *Façade*'s drama manager requires is explained as being provided by an author [28, 33]. Utilities offer a way to influence the way that a narrative AI system will behave, because they are often used by such systems to (i) differentiate between potential alternatives and (ii) infer useful meaning from the narrative world state. *Façade*'s desired tension trajectory is an example of the former, while its method for estimating the story's *current* level of dramatic tension is an example of the latter.

In player-specific experience management (recall Algorithm 1), one or more utilities are used to estimate a player model [43, 46–48]. A *player model* is a mathematical representation of some aspect of a player; this might represent their personality, their knowledge of the story, their preferences over different types of content, their expected emotional reactions, or more. By defining the dimensions of a player model and creating a utility to estimate each of them, an author can influence how the experience manager makes its decisions.

3.2 Altering the System

If an author can gain access to the program code that defines how a narrative AI system works, they might be able to change it to make it work in a different way. In one example, Riedl & Stern took the *ABL* behaviour specification language that was created for *Façade* [28, 49] and combined it with an experience manager based on Narrative Planning called the *Automated Story Director* [50]. Later, Ramirez and Bulitko obtained the source code to the *Automated Story Director* and adapted it to add a player model based on Thue et al.'s *PaSSAGE* [46], creating a player-specific experience manager called *PAST* [51, 52].

Notably, all of the prior works are examples of technically-savvy people acting both as the developers of narrative AI systems *and* as the authors of the IDNs that included those systems. While positive examples exist of diversely skilled teams creating compelling IDNs that rely on narrative AI systems (including *Prom Week* [35, 53], *The Ice-Bound Concordance* [54, 55], *Blood & Laurels* [56, 57], *Nothing for Dinner* [58, 59], and more), finding a way to make system modification more widely accessible to authors remains an open research problem.

With *Mimisbrunnur*, authors were able to preview outputs of its narrative planner and, if desired, mark any outputted plan of action as an unacceptable solution [32][2]. From that point forward, the system would remember and abide by that decision, never showing the marked solution again. While simple in this application, adding a similar capacity for incremental modification to future narrative AI systems might allow authors to alter their operation in a more accessible way.

3.3 Refining or Repurposing a System's Outputs

Throughout this chapter, we have considered authoring as a process of making and acting upon decisions in a narrative context. While narrative AI systems have been used to make a variety of authoring decisions, the task of acting upon those decisions (*e.g.*, making the next dramatic beat actually happen in *Façade*) it is typically left to other parts of the IDN system. This interface that exists between the narrative AI system's output and the remainder of the IDN system presents the last opportunity for an author to influence the decisions that the narrative AI system makes: if each decision can be intercepted and revised or repurposed as desired, the author can gain the benefit of the AI system's operation while still influencing its results. The mode in which an author can do this sort of refinement or repurposing depends on the timing of the narrative AI system's decisions, relative to any player's experience of the larger IDN system's product.

For AI system decisions that are made *before* any player's experience (*e.g.*, to generate a backstory for a character that every player will encounter), choosing

[2] Both *PaSSAGE* [46] and *Mimisbrunnur* [32] are the result of collaborations between this chapter's author and others.

among several of the system's outputs by hand can be a viable option—provided that the system can generate outputs quickly enough to be useful. In this mode of refinement, the system's output serves a starting point for the author's subsequent creative process. When the size of the set of possible outputs is very large, a degree of automation can help. *Story sifting* [60, 61] is a process of searching through a generated sequence of events and identifying subsequences that are salient in some way, and this is typically done on the basis of flexible patterns that one or more authors specify to guide the search.

When AI system decisions are made *during* any player's experience, a more nimble approach is required. For example, given an interactive narrative AI system that simulated the social interactions of several story characters, an author might create one or more utilities that attempt to recognize certain patterns of happenings (*e.g.*, those that involve one character betraying another) and bring the matches to the player's attention. This process is called *incremental story sifting* [62], and it represents some of the newest work in this direction. More about story sifting can be learned in Chap. 'Authoring for Story Sifters'.

4 Summary

From an author's perspective, intelligent narrative technologies are the elements of narrative AI systems, and these systems can share in the task of authoring IDN systems and products. When working with a narrative AI system, it can benefit an author to understand both how the system behaves and what they might do to influence its behaviour. This behaviour can be understood by either experimenting with the system to observe it under different circumstances, or examining its pseudocode or program code to learn about how it works. Meanwhile, an author's opportunities to influence a narrative AI system's decisions come in three forms. First, they can determine its inputs, either by building the collections of content that it uses, setting its parameters, or defining the utilities that it uses as a part of its operation. Second, they can modify how the system itself works by altering its program code, though this avenue presently lacks accessibility for non-technical authors. Third, they can refine or repurpose the output that the AI system produces, potentially by specifying patterns that identify outputs that are of particularly high value. By understanding how a narrative AI system behaves and what they can do to make it better, authors can benefit from the generative capabilities of AI systems while still pursuing the stories they wish to tell.

Acknowledgements The author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), Grant #2020-06502.

References

1. Ryan J (2017) Grimes' fairy tales: a 1960s story generator. In: Nunes N, Oakley I, Nisi V (eds) Interactive storytelling. Springer International Publishing, Cham, pp 89–103
2. Bates J (1992) Virtual reality, art, and entertainment. Presence: J Teleoper Virtual Environ 1(1):133–138
3. Kelso MT, Weyhrauch P, Bates J (1993) Dramatic presence. Presence: J Teleoper Virtual Environ 2(1):1–16
4. Weyhrauch P (1997) Guiding interactive drama. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA
5. Mateas M (1997) An OZ-centric review of interactive drama and believable agents. Carnegie Mellon University
6. Mateas M, Sengers P (eds) (1999) Narrative intelligence—papers from the 1999 fall symposium, vol FS-99-01. AAAI Press
7. Mateas M, Sengers P (eds) (2003) Narrative intelligence. John Benjamins. https://www.jbe-platform.com/content/books/9789027297068
8. Magerko BS, Riedl MO (eds) (2007) Intelligent narrative technologies—papers from the 2007 AAAI fall symposium. AAAI Press
9. Osborn JC (ed) (2020) Joint proceedings of the AIIDE 2020 workshops, co-located. In: CEUR workshop proceedings on 16th AAAI conference on artificial intelligence and interactive digital entertainment (AIIDE 2020), vol 2862. http://ceur-ws.org/Vol-2862/
10. Louchart S, Mehta M, Roberts DL (eds) (2009) Intelligent narrative technologies II—papers from the 2009 AAAI spring symposium, vol SS-09-06. AAAI Press
11. Bahamón J, Fassone R, Mawhorter P, Poulakos S, Robertson J, Ryan J (eds) (2017) Intelligent narrative technologies—papers from the 2017 AIIDE workshop, vol WS-17-20. AAAI Press
12. Barot C, Li B, Rowe J, Tomai E (eds) (2015) Intelligent narrative technologies and social believability in games: papers from the 2015 AIIDE workshop, vol WS-15-22. AAAI Press
13. Cavazza M, Si M, Zook A (eds) (2013) Intelligent narrative technologies: papers from the 2013 AIIDE workshop, vol WS-13-21. AAAI Press
14. Jhala A, Riedl M, Roberts DL (eds) (2010) INT3'10: proceedings of the intelligent narrative technologies III workshop. Association for Computing Machinery, New York, NY, USA
15. Tomai E, Elson D, Rowe J (eds) (2011) Intelligent narrative technologies IV—papers from the 2011 AIIDE workshop, vol WS-11-18. AAAI Press
16. Ware SG, Zhu J, Hodhod R (eds) (2012) Intelligent narrative technologies—papers from the 2012 AIIDE workshop, vol WS-12-14. AAAI Press
17. Wu HY, Si M, Jhala A (eds) (2018) Proceedings of the joint workshop on intelligent narrative technologies and workshop on intelligent cinematography and editing co-located. In: CEUR workshop proceedings on 14th AAAI conference on artificial intelligence and interactive digital entertainment (AIIDE 2018), vol 2321
18. Zhu J, Horswill I, Wardrip-Fruin N (eds) (2014) intelligent narrative technologies 7—papers from the 2014 workshop, vol WS-14-21. AAAI Press
19. Nack F, Gordon AS (eds) (2016) Interactive storytelling. In: 9th international conference on interactive digital storytelling (ICIDS 2016), LNCS, vol 10045. Springer, Cham
20. Finlayson MA, Richards W, Winston PH (2010) Computational models of narrative: review of a workshop. AI Mag 31(2):97. http://orcid.org/10.1609/aimag.v31i2.2295, https://ojs.aaai.org/index.php/aimagazine/article/view/2295
21. Finlayson MA (ed) (2010) Computational models of narrative—papers from the 2010 AAAI fall symposium, vol FS-10-04. AAAI Press
22. Finlayson MA (ed) (2012) The third workshop on computational models of narrative (CMN'12). http://narrative.csail.mit.edu
23. Finlayson MA, Fisseni B, Löwe B, Meister JC (eds) (2013) 2013 workshop on computational models of narrative (CMN 2013), vol 32. OASICS
24. Finlayson MA, Meister JC, Bruneau EG (eds) (2014) 2014 workshop on computational models of narrative (CMN 2014), vol 41. OASICS

25. Finlayson MA, Miller B, Lieto A, Ronfard R (eds) (2015) 6th workshop on computational models of narrative (CMN 2015), vol 45. OASICS
26. Miller B, Lieto A, Ronfard R, Ware SG, Finlayson MA (eds) (2016) 7th workshop on computational models of narrative (CMN 2016), vol 53. OASICS
27. Koenitz H (2010) Towards a theoretical framework for interactive digital narrative. In: Aylett R, Lim MY, Louchart S, Petta P, Riedl M (eds) Interactive storytelling. Springer, Berlin, Heidelberg, pp 176–185
28. Mateas M, Stern A (2007) Façade. http://www.interactivestory.net/
29. Mateas M (2002) Interactive drama, art, and artificial intelligence. PhD thesis, Carnegie Mellon University
30. Thue D (2020) What might an action do? toward a grounded view of actions in interactive storytelling. In: Proceedings of the 13th international conference on interactive digital storytelling (ICIDS'20), LNCS, vol 12497. Springer, Cham, pp 212–220. https://rise.csit.carleton.ca/pubs/Thue_ICIDS_2020.pdf
31. Liapis A, Yannakakis GN, Togelius J (2013) Sentient sketchbook: computer-aided game level authoring. In: Proceedings of the 8th international conference on foundations of digital games, society for the advancement of the science of digital games, pp 213–220
32. Stefnisson I, Thue D (2018) Mimisbrunnur: AI-assisted authoring for interactive storytelling. In: Proceedings of the 14th AAAI conference on artificial intelligence and interactive digital entertainment, AAAI Press, pp 236–242. https://aaai.org/ocs/index.php/AIIDE/AIIDE18/paper/view/18116/17248
33. Mateas M, Stern A (2005) Structuring content in the Façade interactive drama architecture. In: Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment, AAAI Press, pp 93–98, https://ojs.aaai.org/index.php/AIIDE/article/view/18722
34. Latitude (2019) AI dungeon. https://play.aidungeon.io/
35. McCoy J, Treanor M, Samuel B, Reed AA, Mateas M, Fruin NW (2012) Prom week. https://promweek.soe.ucsc.edu/
36. Salge C, Green MC, Canaan R, Togelius J (2018) Generative design in minecraft (GDMC): settlement generation competition. In: Proceedings of the 13th international conference on the foundations of digital games, FDG'18. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3235765.3235814, https://doi.org/10.1145/3235765.3235814
37. Bay 12 Games (2006–2021) Dwarf fortress. http://www.bay12games.com/dwarves/
38. Ludeon Studios (2013) Rimworld. https://rimworldgame.com
39. Riedl MO, Young RM (2010) Narrative planning: balancing plot and character. J Artif Intell Res 39:217–268
40. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H (eds) Advances in neural information processing systems. Curran Associates, Inc., vol 33, pp 1877–1901
41. Author Unknown (2020). https://prompts.aidg.club/
42. Riedl MO, Stern A, Dini D, Alderman J (2008) Dynamic experience management in virtual worlds for entertainment, education, and training. Int Trans Syst Sci Appl Spec Issue Agent Based Syst Hum Learn 4(2):23–42
43. Thue D (2015) Generalized experience management. PhD thesis, University of Alberta, Canada
44. Horswill I (2020) A declarative PCG tool for casual users. In: Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment, vol 16. AAAI Press, pp 81–87
45. Horswill ID, Robison E (2022) Imaginarium—January 2022 release. https://github.com/ianhorswill/Imaginarium
46. Thue D, Bulitko V, Spetch M, Wasylishen E (2007) Interactive storytelling: a player modelling approach. In: 3rd AI and interactive digital entertainment conference (AIIDE 2007). AAAI Press, Palo Alto, California, pp 43–48

47. Sharma M, Mehta M, Ontañón S, Ram A (2007) Player modeling evaluation for interactive fiction. In: Technical report on AIIDE 2007 workshop on optimizing player satisfaction. AAAI Press, Palo Alto, California
48. Thue D, Bulitko V, Spetch M (2008) Making stories player-specific: delayed authoring in interactive storytelling. The first joint international conference on interactive digital storytelling. Springer, Berlin/Heidelberg, Erfurt, Germany, pp 230–241
49. Mateas M, Stern A (2004) A behavior language: joint action and behavioral idioms. In: Prendinger H, Ishizuka M (eds) Life-like characters: tools, affective functions, and applications. Springer, Berlin, Heidelberg, pp 135–161
50. Riedl MO, Stern A (2006) Believable agents and intelligent story adaptation for interactive storytelling. In: 3rd international conference on technologies for interactive digital storytelling and entertainment. Springer, Darmstad, DE, pp 1–12
51. Ramirez AJ, Bulitko V (2012) Telling interactive player-specific stories and planning for it: ASD + PaSSAGE = PAST. In: The eighth artificial intelligence and interactive digital entertainment conference (AIIDE 2012). AAAI Press, Palo Alto, California, pp 173–178
52. Ramirez A, Bulitko V (2015) Automated planning and player modelling for interactive storytelling. IEEE Trans Comput Intell AI Games 7(4):375–386
53. McCoy J, Treanor M, Samuel B, Reed A, Mateas M, Wardrip-Fruin N (2013) Prom week: designing past the game/story dilemma. In: Proceedings of the 8th international conference on the foundations of digital games (FDG 2013), Chania, Crete, Greece
54. Reed A, Garbe J, Apostol N (2013) The ice-bound concordance. www.ice-bound.com
55. Reed A (2017) Changeful tales: design-driven approaches toward more expressive story games. PhD thesis, UC Santa Cruz
56. Evans R, Short E, Nelson G (2014) Blood and laurels. https://versu.com/2014/05/28/blood-laurels/
57. Evans R, Short E (2014) Versu—a simulationist storytelling system. IEEE Trans Comput Intell AI Games 6(2):113–130. https://doi.org/10.1109/TCIAIG.2013.2287297
58. Szilas N, Dumas J, Richle U, Boggini T, Habonneau N (2015) Nothing for dinner. http://nothingfordinner.org/
59. Szilas N (2015) Reconsidering the role of AI in interactive digital narrative. In: Ferri G, Haahr M, Sezen D, Sezen TI, Koenitz H (eds) Interactive digital narrative. Routledge
60. Ryan J (2018) Curating simulated storyworlds. PhD thesis, University of California, Santa Cruz
61. Kreminski M, Dickinson M, Wardrip-Fruin N (2019) Felt: a simple story sifter. In: Proceedings of the 12th international conference on interactive digital storytelling (ICIDS 2019). Springer, Berlin, Heidelberg, pp 267–281
62. Kreminski M, Dickinson M, Mateas M (2021) Winnow: a domain-specific language for incremental story sifting. In: Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment, AAAI Press, pp 156–163. https://ojs.aaai.org/index.php/AIIDE/article/view/18903