

# Iterative Authoring Using Story Generation Feedback: Debugging or Co-creation?

Ivo Swartjes and Mariët Theune

Human Media Interaction group, Twente University  
{swartjes,theune}@ewi.utwente.nl

**Abstract.** We explore the role that story generation feedback may play within the creative process of interactive story authoring. While such feedback is often used as ‘debugging’ information, we explore here a ‘co-creation’ view, in which the outcome of the story generator influences authorial intent. We illustrate an iterative authoring approach in which each iteration consists of idea generation, implementation and simulation. We find that the tension between authorial intent and the partially uncontrollable story generation outcome may be relieved by taking such a co-creation approach.

**Keywords:** Interactive storytelling, authoring, methodology, co-creation.

## 1 Introduction

Creating virtual worlds that are rich in dramatic content and afford users to have a story-like experience within such worlds has proven to be a technically and conceptually challenging goal. Branching narrative techniques are still the state-of-the-art in computer games [1], but are limited in the sense that the combinatorial explosion of possible stories makes it unfeasible to allow more than a few choice points. Tools have been developed that support the authoring of branching narrative [2,3], but for interactive stories that offer true agency to the player, it is unlikely that such tools will be enough to author the huge amount of story content that needs to be available. The approach often pursued in interactive storytelling research to reduce the amount of authoring required, is to use story generation techniques that automatically generate parts of the space of possible story lines.

Using story generation techniques within an interactive storytelling system significantly changes the way that authors can craft their work [4]. It requires a reconception of what the system affords authors to author (which is now no longer the full branching narrative) and of how the content they author relates to the overall space of possible stories (the authorial and interpretive affordances of the system, [5]). It is likely that such an authoring process of *implicit creation* [4] benefits from story generation feedback during authoring. Several authoring tools explicitly provide support for such feedback [4,6,7,8,9]<sup>1</sup>. For instance, the

---

<sup>1</sup> For an overview of authoring tools and their relationship to story generation, see [8].

authoring toolset for the *Madame Bovary* system of Pizzi and Cavazza [8] provides step-by-step plan simulation to allow the author to visualize and modify the space of possible plans that the system generates.

Such feedback may also affect the *creative* process of the author. We identify two views on the relationship between story generation and the creative process. We can view story generation purely as a way to take some of the less creative aspects of authoring out of the hands of the author. We may also ascribe creativity to the story generator, where it is expected to contribute novelty to the story domain. The first view suggests a process that we may call *debugging*: authors adapt the story content or the story generator in such a way that the resulting space of stories matches their unchanging authorial intent. The second view suggests a process of *co-creation*: authors embrace the (sometimes unpredictable and surprising) output of the story generator as a contribution to the space of possible stories, letting it change their initial authorial intent and accepting that the possibilities and limits of these processes take a fundamental part in shaping and constraining the story space. Such a co-creation process can be justified in many cases because – as also noted by [7] – the fictional nature of story worlds allows for approaches in which authoring not only means constructing the domain representation but also the domain itself.

In this paper, we illustrate how a co-creation attitude can reduce tensions between authorial intent and the partially uncontrollable outcome of story generation. The context of our exploration is The Virtual Storyteller, a story generation system based on the emergent narrative concept [10]. Exploring the co-creation view, we describe the authoring process of two small story domains, following an iterative authoring cycle in which feedback was crucial to the conception of the domains. First, we illustrate the attitudes of debugging and co-creation, based on the story generator TALE-SPIN [11]. Then, we show how we incorporate the co-creation approach in an iterative authoring cycle, using our experiences with the two story domains as examples. Finally, we round off the discussion by providing conclusions and future work.

## 2 Mis-spun Tales or Opportunities for Un-told Tales?

To illustrate attitudes of debugging and co-creation, we revisit and discuss in this section two tales generated by one of the first story generation systems, TALE-SPIN by James Meehan [11]. TALE-SPIN is relevant here because its character-centric approach is also the approach of The Virtual Storyteller.

The two tales of Fig. 1 were classified by Meehan as ‘mis-spun’. The first story was mis-spun because Bill did not see Henry Ant. To fix this domain under-representation, Meehan took the debugging approach by introducing ‘noticing’ inferences in TALE-SPIN. The second story had a different reason to be classified as ‘mis-spun’: the authorial intent was to reproduce the Aesop fable “The Fox and the Crow”. Meehan wanted the fox to trick the vain crow into dropping the cheese. From this viewpoint, the hungry crow eating the cheese himself is not only unexpected but also unwanted, as it takes away the opportunity for a

**Story 1**

HENRY ANT WAS THIRSTY. HE WALKED OVER TO THE RIVER BANK WHERE HIS GOOD FRIEND BILL WAS SITTING. HENRY SLIPPED AND FELL IN THE RIVER. HE WAS UNABLE TO CALL FOR HELP. HE DROWNED.

**Story 2**

ONCE UPON A TIME THERE WAS A DISHONEST FOX AND A VAIN CROW. ONE DAY THE CROW WAS SITTING IN HIS TREE HOLDING THE PIECE OF CHEESE. HE BECAME HUNGRY AND SWALLOWED THE CHEESE. THE FOX WALKED OVER TO THE CROW. THE END.

**Fig. 1.** Two ‘mis-spun’ tales generated by Meehan’s story generator TALE-SPIN

story about deception by the fox. Meehan again took the debugging approach: he changed the setup to match his authorial intent by making sure that the crow was not hungry, so that the actual Aesop fable would emerge.

We can imagine though, that only taking the debugging approach is a rather brittle way to author content for a story generation concept such as that of TALE-SPIN. The more content is authored, the more likely it becomes that the emergence of the deception story is disrupted due to unexpected interactions. In contrast, taking the co-creation approach would mean that possible variations, resulting from a nondeterministic simulation of characters, are accepted as opportunities. Story content is authored for the continuation or improvement of these variations. Noticing that it is possible for the crow to eat the cheese because he is hungry, Meehan could have embraced this unexpected possibility, letting it inspire subsequent authoring. For instance, he might have made the cheese actually belong to the fox. He could then have authored the possibility for the fox to become angry with the crow eating his cheese, leading to a revenge plan. Such a co-creation approach, in which authorial intent is affected by the outcome of the story generator, is the focus of our exploration of authoring for The Virtual Storyteller. The following sections describe and illustrate how we incorporate this approach in an iterative authoring process.

### 3 Authoring with the Virtual Storyteller

The Virtual Storyteller is a multi-agent story generation framework based on the *emergent narrative* approach [12] which has much in common with TALE-SPIN in the sense that stories emerge from character behaviour.

At run time, the roles of characters in the story world are played by agents using a BDI-like mental model to govern action selection. Actions of one character may affect other characters, resulting in the production of a network of temporally and causally related story elements. These story elements are story world setting facts, character goals, actions of characters, outcomes of goal pursuit, perceptions, cognitive states (beliefs, emotions, etc.) and (unintentional or external) events. This formalism, based on story comprehension theory, is described in [13]. The outcomes of the cognitive processes of the virtual characters

are stored as causal links. For instance, a planning algorithm determines how character goals motivate actions (stored as “goal A motivates action B” causal links), and a goal management process determines how cognitive states cause the adoption of new goals (stored as “emotion X psychologically causes goal Y” causal links). These causal links yield a network of causally related elements, forming a representation of the fabula of a story to be narrated afterwards.

The Virtual Storyteller aims to avoid the global coherence problem that TALE-SPIN suffered from by giving the characters a certain actor-awareness. As actors of their own stories, the agents use out-of-character processes such as *late commitment*, which means that they fill in relevant details of the story world at simulation time using *framing operators* [14], and *event planning*, which means that they play god by selecting events to happen. These processes give the characters flexibility to control the emerging event sequence, by further defining and influencing the story world, aiming for more causal connectedness of the emerging event sequence. For instance, as actors they can actively satisfy the preconditions of a goal, controlling when it is adopted. We call this process *goal justification*. An example from a pirates story domain: a goal to refill the water supply of a ship might have the preconditions that the ship has a water supply, that this water supply is exhausted, and that the character adopting the goal is the captain of the ship. Framing operators with the meaning “Let’s pretend the ship has a water supply” and “Let’s pretend I am the captain of this ship”, and an external event with the meaning “The water supply became exhausted” can satisfy any of these preconditions if they were not fulfilled yet.

### 3.1 An Authorial Impasse

We consider the causal network formalism to be independent of any particular story domain, and inherent to the Virtual Storyteller system. The author of a story domain is concerned with the questions: (1) which instances of story elements are there in the domain: which actions, goals, etc. may occur? (2) how do they causally connect: when do these element instances occur, in causal relation to others? For instance, how will a pirate captain select his actions based on his internal state? How does a little girl respond emotionally to a scary wolf? And a grandmother? It might be useful to think of these two authoring concerns as *content authoring* and *process authoring*, respectively, although the distinction is somewhat artificial. It means that authors using this concept are required to also have cognitive modelling and programming skills. Content and process authoring constitute the implementation of a ‘landscape of possible stories’ when the system is run (the *story landscape*, [15]). Our focus here is not on the specific technical solutions for content and process authoring that were found for The Virtual Storyteller, but on the creative process by which one may develop such a story landscape.

The task of constructing a story landscape presents the author with a certain impasse. There is no direct architectural support for story lines in The Virtual Storyteller, as these are generated at run time as an effect of content and process authoring. On the one hand, the TALE-SPIN example illustrated that having a strong concern for the emergence of specific story lines leads to an approach in which content and process authoring are done in such a way that the intended

stories are produced. This is not only difficult, as it requires making predictions of the interaction of the behaviours of the character models and settings, but is also bound to result in a frustrating mismatch between what story lines one wants and what story lines emerge. On the other hand, we found that having *no* concern for desired story lines is also problematic, because it leads to a style of authoring in which arbitrary choices must be made in content and process authoring<sup>2</sup>. The authorial impasse is that writing characters also means considering the role they play in the emergence of story lines.

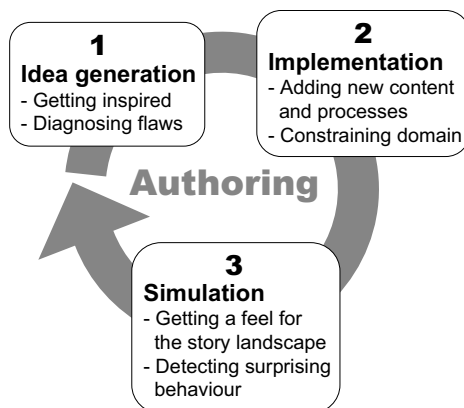
### 3.2 An Iterative Authoring Cycle

We found that a good way to resolve this impasse is to develop virtual characters and the story lines they take part in simultaneously, thereby authoring the story landscape in an iterative authoring cycle, as also suggested by [16, pp.154-157]. Here we elaborate on the authoring cycle by incorporating the notion of co-creation. See Fig. 2. An initial bootstrap design of a story world forms the start of an authoring process in which the author’s intent for the story world is meant to be still vague or absent, but becomes clearer as authoring progresses:

1. **Idea generation.** The story landscape, laid out by the content and processes authored so far, inspires authorial ideas for how it can be extended. These ideas might be directly implementable as story world content (content authoring), or might require extending the cognitive processes of the characters (process authoring). The simulation may also have displayed undesired behaviour (e.g., the lack of ‘noticing’ inferences in TALE-SPIN), which must be diagnosed (is it a domain underrepresentation? A flaw in the AI? Or simply an uninspiring development?) and treated. The choice between these two approaches (co-creation or debugging) is up to the author.
2. **Implementation.** Based on the new ideas, story content is added to the system (content authoring), and new cognitive processes are implemented (process authoring). Based on the observation of undesired behaviour of the simulation, flaws in the representation are repaired, or the domain is further constrained. The authored content and processes may expand and change the story landscape in several ways, some of which were not directly intended by the author or even surprising, due to the complex interaction of the authored material.
3. **Simulation.** Running simulations of the story world under development give the author a feel for the current shape of the story landscape, and a feel for what the system can do with the authored material so far. It also exposes unintended and surprising effects of the authored material, which may inspire the author in a subsequent cycle.

---

<sup>2</sup> In early development phases of The Virtual Storyteller, we attempted to build ‘a general purpose knowledge representation for story worlds’. We found that we often justified choices based on vague, general notions of authorial intent. For instance, we chose to include a character’s health, because in many stories, characters become sick. Without the context of a specific story world, such a choice is arbitrary.



**Fig. 2.** Iterative authoring cycle for emergent narrative. A story landscape is formed iteratively, each iteration consisting of three consecutive phases: (1) idea generation, (2) implementation, and (3) simulation.

Within this authoring cycle, choices between debugging and co-creation must be made continuously. For example, we once modelled a goal for a pirate character to shoot an enemy ship, and two actions: to load a cannon and to fire at the ship. This had an unexpected effect on the story landscape: a pirate fired himself from the cannon to get to a nearby island. This was certainly not intended, but is it “wrong”? We should realize that this is an authorial choice. If we take the debugging approach (saying it does not match our authorial intent), then the action to fire something from a cannon must be further constrained to specify that only cannon balls can be fired. If we take the co-creation approach (adapting our authorial intent), then we incorporate this behaviour into the story domain. We proceed, for instance, by modelling that shooting oneself from a cannon hurts, and that a pirate, as a consequence, will use the cannon only in emergency situations (e.g., to escape from a fight), and with reluctance.

The latter approach builds on the idea in improvisational theatre that there is no “wrong” direction to an improvised story; anything can serve as a basis to further the story if properly incorporated. Improv actors start a scene with a trivial beginning, without knowing where it will take them, and then use association and knowledge of story progression to make authorial choices in context of what emerges. In the case of the proposed authoring cycle, it is the author who makes authorial choices ‘in the moment’, by simulating the system, deciding what might happen in certain situations, and implementing such decisions.

## 4 Two Case Studies

This section discusses two small story domains that were authored with The Virtual Storyteller: a domain about pirates, and a domain based on the “Little Red Riding Hood” story. Note that these domains are discussed with a focus on

the authoring process, rather than on the precise formalisations that underlie the authored material, which are detailed elsewhere [10].

#### 4.1 Pirates

The Pirates story domain `PLUNDER`<sup>3</sup> is the main domain that the Virtual Storyteller project is focusing its efforts on. Pirates portray stereotypical behaviour that is somewhat easier to model than, say, the behaviour of characters in psychological thrillers. This property is shared with the domain of fairy tales, but unlike fairy tales, we believe that pirate tales evoke less expectations of story structure for the reader (e.g., adherence to Propp’s morphology).

Rather than a fully generated plot, `PLUNDER` has an overarching storyline (a crew of pirates going on a treasure hunt) that forms the theme for several story subdomains. We brainstormed about the following questions: (1) what are some subdomains that we can conceive of to start building simulations around? (2) what are typical roles of pirates in such subdomains? (e.g., captain, boatswain) (3) what do pirates want and do there? (e.g., drink rum, commit mutiny) (4) what are typical objects associated with these subdomains? (e.g., ship, treasure map, pirate flag). We considered subdomains in which simple stories would take place: in a bar where a crew is assembled, on a ship underway to a treasure island, and on the island where the treasure is buried.

Our initial, naive authoring approach was to enforce a co-creation process by creating a ‘big bag of domain content’ to bootstrap the authoring iterations, and to see if we could use it as a basis for subsequent iterations. For the process authoring side of the domain, we reused domain-independent components developed earlier, such as a partial-order planner that allows characters to select actions based on their goals. The first subdomain we aimed to develop was called *on the ship*. We created two pirate characters: Anne Bonney and Billy Bones. We implemented a few goals that these pirates might have aboard a ship, and added some actions that allowed the characters to construct goal plans. For goal justification, we also added some events and framing operators. See Table 1.

*Cycle 1.* In an example simulation run, Anne Bonney used a framing operator to endow herself to be the ship’s captain, and selected an event that the water supply became exhausted. This matched the preconditions of the goal to replenish it, that she subsequently adopted. Billy Bones selected an event to become thirsty for some rum, and used a framing operator to establish the fact that there was rum available in the hold of the ship. This justified the adoption of the goal to drink rum. Anne Bonney sailed to a nearby island, went ashore and started replenishing the water supply, whilst Billy Bones opened the hatch to the hold, went to the hold, picked up a bottle of rum and got himself drunk. After this, Billy Bones framed himself to hate Anne Bonney and adopted the goal to kill her. Planning how to do this, he endowed himself with a rapier and went ashore to find and stab Anne.

---

<sup>3</sup> Pirates Looming in Unscripted Narrative: towards Dramatic Emergent Roleplay.

We observed a certain incoherence in the event sequences. The causal network theory of story comprehension, on which we base our fabula formalism, relates narrative coherence directly to the causal connectedness of the events [17]. It follows that coherence would be improved if the events were better causally connected, e.g., when actions of one goal lead up to the adoption of another goal, or cause a conflict between goals and attitudes. Yet, Billy Bones' goal to kill Anne Bonney had no such causal connection to earlier story elements, including his own goal of getting drunk. We diagnosed this as a flaw in the AI, which we solved by further constraining the goal adoption process of the characters, so that they would only adopt goals that are a result of earlier events. After this debugging step, we took the co-creation view, adding more content to allow for more causal connections whilst building on the goals already present. So far, drinking rum, replenishing the water supply and killing a fellow pirate had little to do with each other. So, we ended up somewhat artificially adding possibilities for causal connections. For instance, we made it possible for a character to hate drunken pirates (not very pirate-like!), leading to murder. Another somewhat artificial addition for causal connection between goals was to add a goal to drink from the water supply (rather than to drink rum), so it would become exhausted.

*Cycle 2.* The newly added content resulted in the expected causal connections. For instance, Anne Bonney got drunk, after which Billy Bones saw her drunk, hated her, and decided to pick a fight. In another run, Billy Bones drunk the last of the water supply so that the goal to go and refill it was triggered for the pirate captain Anne.

Although we were on track towards the emergence of more coherent narratives, we felt that the initial arbitrary choices in story content required nontrivial work in order to embed the content within a larger causal context. It seems

**Table 1.** Operators for bootstrapping the Pirates subdomain *on the ship* (informal representation)

Goal	Actions	Framing operators	Events
<b>Drink rum</b>	<ul style="list-style-type: none"> <li>– Open hatch</li> <li>– Go to hold</li> <li>– Pick up rum</li> </ul>	<ul style="list-style-type: none"> <li>– Rum bottle is in hold</li> </ul>	<ul style="list-style-type: none"> <li>– Become thirsty</li> </ul>
<b>Refill water supply</b>	<ul style="list-style-type: none"> <li>– Sail to island</li> <li>– Go ashore</li> <li>– Replenish water supply</li> </ul>	<ul style="list-style-type: none"> <li>– Pirate is the captain</li> <li>– Ship has a water supply</li> </ul>	<ul style="list-style-type: none"> <li>– Water supply becomes exhausted</li> </ul>
<b>Kill a fellow pirate</b>	<ul style="list-style-type: none"> <li>– Go to pirate</li> <li>– Draw rapier</li> <li>– Stab pirate</li> </ul>	<ul style="list-style-type: none"> <li>– Pirate hates someone</li> <li>– Pirate has a rapier</li> </ul>	



that this issue was at least partially caused by the ‘big bag of domain content’ approach. We concluded that a process of co-creation might benefit from a tighter integration of authoring and feedback, bootstrapping the authoring process from minimal content.

## 4.2 Little Red Riding Hood

The story domain RED (Red’s Emergent Drama) was developed for participation in the *Little Red Riding Hood Panel: The Authoring Process in Interactive Storytelling* at the ICIDS 2008 conference. The co-creation view meant that we immediately let go of the goal to reproduce the original Little Red Riding Hood story, using it only as inspiration for the characters and their interactions.

We used three character agents playing the roles of Little Red Riding Hood (Red), Grandma, and Wolf. To enable a tighter feedback loop between authoring and story generation, we did not bootstrap the authoring process with a ‘big bag of domain content’ this time. Rather, we started with a simple and minimal design of a story world (e.g., one goal, actions that can achieve it, and a planner that turns these actions into a goal plan). In the case of the RED story world, this meant that the initial setup of the story world contained the goal (and accompanying actions) to bring a cake to Grandma. To enable the expression of this goal, a small story world geography was authored (a path from Red’s house to the forest, and from the forest to Grandma’s house) along with some actions (to skip from one location to another, and to give something to someone). With these initial ideas and their implementation, we started the iterative authoring cycle of Fig. 2.

*Cycle 1.* A predictable story ‘emerged’ during simulation: Red skipped to the forest, then to Grandma’s house, and gave her the cake. We chose a continuation in which Grandma would eat the cake, and also wanted to enable Wolf to eat the cake, by stealing it. So we added a goal to eat something, and an action to take something from someone else. To enable goal justification, we added an event to become hungry.

*Cycle 2.* In the simulation, both Grandma and Wolf could not adopt any goals: for them, the preconditions for available goals did not match the initial state of the story world. So they attempted to justify goals (out-of-character) by selecting the event to become hungry. On her way to Grandma, Red met Wolf, who had adopted the goal to eat something and took the cake from Red. However, since Red still wanted to bring the cake to Grandma, she took back the cake from Wolf. Wolf still wanted to eat the cake, so again took the cake from Red. This ‘cake fight’ continued until Red skipped to Grandma’s house before Wolf managed to take the cake from her. Then, a similar situation occurred with Grandma who did not know that Red was going to give her the cake. Because she was hungry, Grandma took the cake from Red. This also happened to solve Red’s goal that Grandma had the cake. Now Red had no more goals to adopt, so she selected

the event to become hungry and adopted the goal to eat something. This caused Red to take back the cake from Grandma, and eat it herself.

Some of this behaviour was expected, some was surprising and inspiring (e.g., Red could be an assertive girl), and some was undesired, resulting from a domain underspecification: only mean persons take something from someone when it does not belong to them; nice people ask. Furthermore, if the cake can be taken away from Red without the possibility of taking it back (e.g., if Red is not mean), a response is needed to this event. We chose to have Red cry as a plausible dramatic response for little girls.

So in the implementation phase, we added a framing operator that can endow a character to be mean; only one character in the story world can be mean (authorial choice). We added a cry reactive action, triggered by someone taking something from a character without the character's consent. Preemptively, we constrained the crying reactive action trigger so that it is only applicable to little girls (authorial choice).

*Cycle 3.* In the simulation, Wolf framed himself to be mean (out-of-character, using the framing operator), so that he could plan to take away the cake from Red. After this mean action from Wolf, Red started crying. This was as expected. As authors, we considered how the simulation might continue at this point. Red might go to Grandma to seek support. In revenge, Grandma might poison a cake and feed it to Wolf. We considered how this might open up possibilities in the simulation for the cake to be poisoned by Red in an attempt to poison Grandma in case *Red* happens to be the mean one (an idea was to add a goal specifying that mean characters try to poison others). Wolf might be allowed to satisfy his hunger by eating Grandma, or by following Red and eating them both. Red might be given the option to be distrustful and avoid interaction with Wolf. We only implemented one of these ideas. We added a goal to seek support and added speech actions for Red to tell Grandma what happened, and to ask her what to do. We also added a goal for Grandma to avenge her granddaughter by poisoning the wrongdoer, and actions that allowed a goal plan.

By authoring like this, authorial choices are always made in a larger story context, avoiding problems of ending up with incoherent story content. Still, to avoid authoring a linear story, we found it necessary to actively consider the ways in which authored material might also be used in different simulation runs (e.g., poisoning as an act of meanness). This is a way to increase density of the story landscape, as discussed in [15].

## 5 Conclusions and Future Work

Using story generation within an interactive storytelling system may reduce the amount of story content that needs to be authored explicitly, but raises the question what the role of story generation is within the creative process of the author. A distinction between a process of debugging (author with an unchanging authorial intent) and co-creation (letting story generation and authorial intent

influence each other) was found useful in order to describe two different ways that story generation feedback during authoring may serve an author.

One important consequence of using the co-creation view for building interactive storytelling systems is that it moves away from the idea of having totally pre-meditated authorial control over the end result. This makes an iterative authoring process essential and is likely to have further consequences for the requirements of authoring tools for such systems. The benefit is that the tension between authorial control and the sometimes unpredictable or uncontrollable story generation outcome is lessened. Another benefit of the co-creation view is that it is suitable for multi-party authoring approaches. A recent example is the massively collaborative authoring approach of [9], in which intelligent story characters can be taught in a rehearsal mode how (not) to act and why.

For *The Virtual Storyteller*, we found that taking the co-creation view using an iterative authoring cycle allows for a flexible incremental approach to story world authoring. Such an approach reduces the authorial impasse that the author of an emergent narrative faces. We have first used the iterative authoring cycle for one of the subdomains of *PLUNDER*, and used a further refined approach in the *RED* domain. We found that bootstrapping the authoring cycle with minimal story world content, followed by small authoring cycles, helps in achieving coherence and density of the story landscape. Based on this refined version, we aim to develop more subdomains of the *PLUNDER* domain.

So far, the developed domains have been rather small in scale. An open question is how the authoring approach proposed here scales up. We foresee two factors that may complicate upscaling. First, since the simulation is nondeterministic, there is no systematic way to explore all possible simulation runs and the unexpected effects that result from any implementation step. Second, the authoring approach is meant to be additive, but might in practice require revisiting earlier choices. Since process and content authoring at any point in the authoring process affect the outcome of the simulation in fundamental ways, it may be possible that authoring choices made early in the authoring process can be undermined in subsequent authoring iterations, especially where these cannot be detected from the logical representation. These are issues that need further exploration.

**Acknowledgements.** This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

## References

1. Costikyan, G.: Games, storytelling, and breaking the string. In: Harrigan, P., Wardrip-Fruin, N. (eds.) *Second Person: Roleplaying and Story in Playable Media*, pp. 5–14. MIT University Press, Cambridge (2007)
2. Silverman, B., Johns, M., Weaver, R., Mosley, J.: Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into Interactive Dramas. In: Balet, O., Subsol, G., Torguet, P. (eds.) *ICVS 2003. LNCS*, vol. 2897, pp. 65–73. Springer, Heidelberg (2003)

3. Sauer, S., Osswald, K., Wielemans, X., Stifter, M.: U-Create: Creative Authoring Tools for Edutainment Applications. In: Göbel, S., Malkewitz, R., Iurgel, I. (eds.) TIDSE 2006. LNCS, vol. 4326, pp. 163–168. Springer, Heidelberg (2006)
4. Spierling, U.: Adding Aspects of “Implicit Creation” to the Authoring Process in Interactive Storytelling. In: Cavazza, M., Donikian, S. (eds.) ICVS-VirtStory 2007. LNCS, vol. 4871, pp. 13–25. Springer, Heidelberg (2007)
5. Mateas, M.: Expressive AI. *Leonardo: Journal of the International Society for Arts, Sciences, and Technology* 34(2), 147–153 (2001)
6. Carbonaro, M., Cutumisu, M., McNaughton, M., Onuczko, C., Roy, T., Schaefer, J., Szafron, D., Gillis, S., Kratchmer, S.: Interactive Story Writing in the Classroom: Using Computer Games. In: Proceedings of DiGRA 2005 Conference: Changing Views – Worlds in Play, pp. 323–338 (2005)
7. Thomas, J.M., Young, M.R.: Author in the Loop: Using Mixed-Initiative Planning to Improve Interactive Narrative. In: Proceedings of the ICAPS 2006 Workshop on AI Planning for Computer Games and Synthetic Characters (2006)
8. Pizzi, D., Cavazza, M.: From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 285–296. Springer, Heidelberg (2008)
9. Kriegel, M., Aylett, R.: Emergent Narrative as a Novel Framework for Massively Collaborative Authoring. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 73–80. Springer, Heidelberg (2008)
10. Swartjes, I., Theune, M.: The Virtual Storyteller: Story Generation by Simulation. In: Nijholt, A., Pantic, M., Poel, M., Hondorp, H. (eds.) BNAIC 2008, Proceedings 20th Belgian-Netherlands Conference on Artificial Intelligence, pp. 257–264. University of Twente, Enschede (2008)
11. Meehan, J.: TALE-SPIN. In: Schank, R.C., Riesbeck, C.K. (eds.) *Inside Computer Understanding: Five Programs Plus Miniatures*, pp. 197–226. Lawrence Erlbaum Associates, Hillsdale (1981)
12. Aylett, R.: Emergent Narrative, Social Immersion and “Storification”. In: Proceedings of the 1st International Workshop on Narrative Interaction for Learning Environments (NILE), Edinburgh (2000)
13. Swartjes, I., Theune, M.: A Fabula Model for Emergent Narrative. In: Göbel, S., Malkewitz, R., Iurgel, I. (eds.) TIDSE 2006. LNCS, vol. 4326, pp. 49–60. Springer, Heidelberg (2006)
14. Swartjes, I., Kruizinga, E., Theune, M.: Let’s Pretend I Had a Sword: Late Commitment in Emergent Narrative. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 264–267. Springer, Heidelberg (2008)
15. Louchart, S., Swartjes, I., Kriegel, M., Aylett, R.: Purposeful Authoring for Emergent Narrative. In: Spierling, U., Szilas, N. (eds.) ICIDS 2008. LNCS, vol. 5334, pp. 273–284. Springer, Heidelberg (2008)
16. Louchart, S.: Emergent Narrative - Towards a Narrative Theory of Virtual Reality. Ph.D. thesis, University of Salford, Salford, UK (2007)
17. Trabasso, T., Secco, T., van den Broek, P.: Causal Cohesion and Story Coherence. In: *Learning and Comprehension of Text*, pp. 83–111. Lawrence Erlbaum Associates, Hillsdale (1982)