

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/252371559>

# The Virtual Storyteller: Story Generation by Simulation

Article · January 2008

CITATIONS

21

READS

4,051

2 authors:



**Ivo Swartjes**

Ranj Serious Games

18 PUBLICATIONS 319 CITATIONS

[SEE PROFILE](#)



**Mariet Theune**

University of Twente

163 PUBLICATIONS 1,807 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ARIA VALUSPA [View project](#)



The Virtual Storyteller [View project](#)

# The Virtual Storyteller: Story Generation by Simulation

Ivo Swartjes

Mariët Theune

*University of Twente, P.O.Box 217, 7500 AE Enschede*

## Abstract

The Virtual Storyteller is a multi-agent framework that generates stories based on a concept called emergent narrative. In this paper, we describe the motivation and approach of the Virtual Storyteller, and give an overview of the computational processes involved in the story generation process. We also discuss some of the challenges posed by our chosen approach.

## 1 Introduction

The Virtual Storyteller is a multi-agent framework for generating stories [7].<sup>1</sup> These stories emerge from a simulation of virtual characters in a story world. Story generation happens in two phases: (1) simulation and (2) presentation (see figure 1). In the simulation phase, Character Agents work together with a Plot Agent to produce an interesting event sequence. The Character Agents simulate a character's life in the story world. They pursue goals, reason about their perceptions, experience emotions and make decisions in the context of this world. In order to constrain the authoring of engaging characters to a manageable degree, we focus on a particular domain of pirate stories. The Plot Agent facilitates the simulation by starting up scenes that specify the initial state of characters and story world for the particular scene. The event sequence resulting from the interaction of the Character Agents is captured by the Plot Agent in a formal representation (the fabula) that forms the input for the presentation phase. The World Agent manages a knowledge representation of the virtual world, executing actions and events as they occur, and sending back the results to the Plot Agent.

In the presentation phase, the Narrator component turns the formal fabula representation into an actual story by selecting the content to tell from the fabula, and applying language generation techniques to it in order to produce a text [8]. Optionally, the Narrator can be coupled to a talking face that tells the story using speech synthesis, thus simulating a human storyteller [2].

The rest of the paper is organized as follows. Section 2 gives a motivation of our work. Section 3 describes the theoretical foundation of our story generation approach. Sections 4 and 5 then describe the technical aspects of the simulation and presentation phase of the Virtual Storyteller, respectively. Finally, section 6 provides some conclusions and future work.

## 2 Motivation

With the advent of virtual reality and progressing computer game development, the computer as a medium is opened up more and more to the possibility of new forms of storytelling. Different from traditional media, the computer affords *interactivity*, adding a new dimension to the experience of stories. The interactor can potentially play an active role in storytelling by becoming a *participant*, determining the course of the story she takes part in. That this is a promising venture, can be seen by the fact that the computer games industry attempts to reconcile story with game play on a regular basis. However, how to do this well from a design perspective remains a difficult question. The reason for this is that there is a problematic tension between interactivity and story. This tension is sometimes called the *narrative paradox*: offering the interactor choices

---

<sup>1</sup>The Virtual Storyteller is implemented in Java on top of the JADE agent framework, using SWI-Prolog for its planning and reasoning processes.

breaks right through the notion of carefully orchestrated plot [1]. Authoring a customized continuation of the plot for all possible interactions quickly leads to an ‘explosion of endings’ and is untenable when the number of interactions increases.

Similar issues arise when interactive storytelling is used for educational purposes, such as scenario-based training. Here, educational goals are translated into a narrative environment that provides the interactor the chance to practice with problematic situations and explore the consequences of possible choices to make. Again, the system has to offer many different outcomes based on the choices of the interactors.

The narrative paradox is one of the fundamental issues that the Interactive Storytelling field is trying to address. One particular approach is offered through the concept of *emergent narrative* [1]. Emergent narrative is a simulation-based approach characterized by the fact that there is no predetermined plot. Instead, stories emerge through the interactions of virtual characters. There is also no longer one story; each character experiences his or her own story.

In the Virtual Storyteller project, we investigate under which conditions stories emerge through character interactions, with the characters being played by intelligent agents. An assumption we make is that we can later replace one of the characters in this setup by creating an interface for a human interactor without having to change much in the rest of the architecture.

### 3 Theory

For the generation of stories, our starting point is to look at how people comprehend stories. The reason to do this is threefold. First, story comprehension theory poses implications for system design because it offers narrative structures to aim for, informing amongst others the design of the cognitive processes of the characters. Second, the Virtual Storyteller can model its own understanding of the emerging story in order to make informed decisions on how it can develop further in the simulation phase. Third, the Virtual Storyteller can make assumptions about how the reader will comprehend the stories it produces, which informs decisions in the presentation phase.

There has been considerable research carried out on the subject of story understanding. Particularly influential research is described in [9], presenting a model that can be used to classify the clauses of a story into a small set of elements, and describe their causal connections. The model subsumes several other story understanding theories (e.g., story grammar and script theories). We use the theory to model the fabula of the simulation (i.e., what happened in the simulation and why). Our fabula model defines causal relationships between seven types of elements: setting elements, goals, actions, outcomes, events, perceptions and internal elements (see figure 2). A *setting element* describes an aspect of the story world that all the characters share and consider to be true. A *goal* is the main drive for a character to act. A goal in this context describes a desire to attain, maintain, leave or avoid certain states, activities or objects [9]. The state of the story world is changed through *actions* and *events*. The difference is that actions are performed intentionally by a character, whilst events are not; they just happen. An *outcome* is a mental concept that relates a goal to its fulfillment or failure. When a character believes that one of its goals is fulfilled, the goal has a positive outcome, but if the character believes that the performed actions did not succeed in fulfilling the goal, the

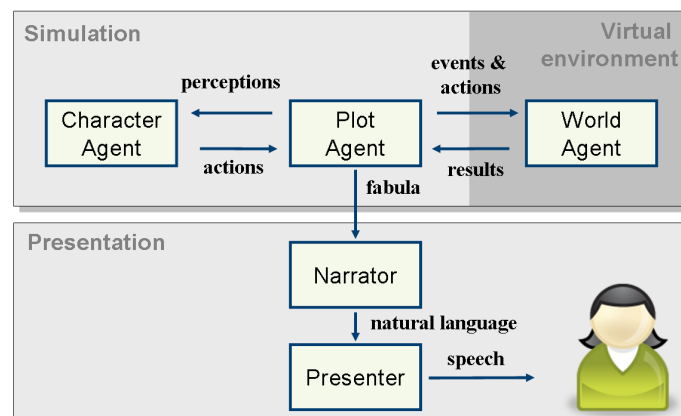


Figure 1: Global architecture of The Virtual Storyteller

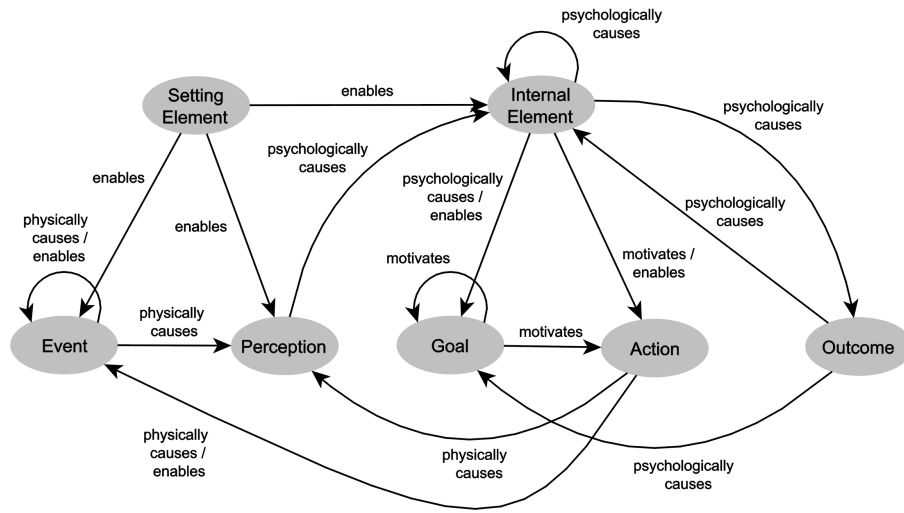


Figure 2: Fabula Model

outcome is negative. A *perception* describes properties of the world that a character witnesses. Everything else that goes on within a character, such as cognitions, emotions, feelings and beliefs, is classified as an *internal element*.

In a fabula, these elements are connected by four types of causalities: *physical and psychological causality* represent unintentional causes of either physical or mental nature, *motivation* represents an intentional cause and *enablement* represents any cause where the causing element is making it possible for the resulting element to occur.

For illustration purposes, figure 3 shows a small example story produced by the Virtual Storyteller using a simple simulation with one character.<sup>2</sup> The following sections explain how stories like this one are produced. Underlying this particular example story are actions such as opening the hatch, sailing to an island, and filling a water supply, motivated by a goal to refill the water supply and enabled by setting elements and beliefs about the existence and location of the island, and a pond on the island. The goal to refill the water supply is psychologically caused by a perception that the water supply is empty, which is physically caused by the event of the water supply running out of water.

*Once upon a time there was a pirate, who was called Billy Bones. He was in the hold of his ship. The water supply was empty and he wanted to fill it. Therefore he opened the hatch. With a ladder the pirate walked to the deck. With the ship he sailed to an island. After he had gone ashore at the island, he filled the water supply with water from a pond.*

Figure 3: Example story produced by The Virtual Storyteller (translation)

## 4 Simulation

At the basis of story production lies the simulation of the story world. The aim of the simulation is to produce a fabula that contains interesting event sequences for the presentation phase. The fabula production is done by modeling virtual characters whose interactions with each other and the story world determine how the story develops. The simulation is based on a multi-agent system in which Character Agents play out the role of the characters they represent, and share responsibility over plot progression. The design of these Character Agents is informed by improvisational theater [6].

The simulation is divided into rounds. Each round, the Plot Agent takes the initiative and requests each Character Agent to select an action. In response, each Character Agent goes through a deliberation

<sup>2</sup>The Narrator produces Dutch text. The original text is: *Er was eens een piraat, die Billy Bones heette. Hij was in het ruim van zijn schip. De watervoorraad was op en hij wilde hem vullen. Daarom opende hij het luik. Met een ladder liep de piraat naar het dek. Met het schip voer hij naar een eiland. Nadat hij bij het eiland aan land was gegaan, vulde hij de watervoorraad met water uit een vijver.*

cycle and responds with an action it wants to perform, or refuses the request, for instance because it is already performing an action, or because it cannot think of any action. When an action finishes, its effects are translated into perceptions that each Character Agent receives. The Character Agent is also informed whether an action was successful or failed.

## 4.1 Late commitment

One of the difficulties with the use of simulation for the emergence of stories is that the emerging stories are very much dependent on the initial setup of the story world, i.e., which characters, objects and relationships are defined. It is difficult for an author to try to imagine the effects of the way she sets up the story world in terms of the stories that come out and yet, informed decisions must be made. For instance, should there be a crate of rum bottles in the hold? Should two pirates hate each other? Or distrust each other? Is the ship they are on already carrying lots of gold, or is it empty? These kind of details might come to play an important role in a particular story, or they might not.

What makes the Virtual Storyteller unique in this respect, is that it is designed so that these details can be filled in as they become useful for the story under development, so that the initial setup can be kept small and basic. Based on the observation that actors in improvisational theater also do not work with a predetermined, agreed upon story world, but frame this world as they go along [5], we have developed a technique that enables the use of this kind of decisions using operators that we call *framing operators*. Framing operators are a special kind of operators; they add new knowledge to the state of the story world (not only the physical state such as the objects and locations, but also the social state such as properties and relationships of characters) whilst creating the illusion that the knowledge has already always been present. In effect, framing operators implement a way for the character agent to say for instance: “Let’s pretend I was the captain of the ship,” or “Let’s pretend I couldn’t swim”. This takes away the burden of having to determine the complete state of the story world in advance, and allows the agents to fill in particular details when useful. For instance, if it comes to be that the pirate ship is attacked by another ship, it is interesting if it is loaded with gold and the Virtual Storyteller can introduce this fact at the appropriate time. The decision to use framing operators is currently deeply interwoven with the action planning and goal selection processes of the characters. In the fabula, the effects of an executed framing operator end up as setting elements.

For instance, the starting point for the example story of figure 3 was Billy Bones’ use of late commitment to justify the adoption of the goal to refill the water supply. This goal has preconditions stating that the character adopting the goal is the ship’s captain (since captains carry this responsibility), that the ship has a water supply, and that the water supply is empty. Planning to establish this context, the Character Agent has used two framing operators (`BeCaptain` and `ShipHasWaterSupply`) and an event (`OutOfWater`). Figure 4 shows the `BeCaptain` framing operator in more detail.

We have also addressed some of the implications the use of late commitment has for the consistency of the simulation. All characters should accept the effects of a framing operator and be able to pretend they have always been true, so the characters should not treat the information as if it were new by for instance responding emotionally. To this end, we have implemented a mechanism to check if all characters can consent to the execution of a framing operator whenever selected by a character.

BeCaptain	
<i>Preconditions:</i> <code>ship(?ship)</code> <code>character(?me)</code> <code>¬ captain(?me)</code> <code>¬ ( captain(?other)</code> <code>    ∧ owns(?other, ?ship) )</code>	<i>Effects:</i> <code>captain(?me)</code> <code>owns(?me, ?ship)</code>

Figure 4: The `BeCaptain` framing operator. Whenever there is a ship, and a character that is not a captain (of any ship) yet, and the ship does not have a captain already, the `BeCaptain` framing operator can define that the character is the captain of the ship and owns it.

## 4.2 Domain modelling

The domain of a particular story world is made up by representing the *setting* of the story world (the initial state of the simulation), and providing *schemas* that contain process-based information used by the cognitive processes of the characters.

The setting is based on an ontology describing the particular concepts of the story world. For our pirates world, this ontology contains concepts like *Sword* and *Pirate*. It also defines relationships relevant to the domain, such as *hates* and *owns*. In the setting, we use these concepts and relationships to represent the knowledge about the story world as it is in the beginning of a scene.

The schemas are data structures representing the actions that characters can perform, but also external events, character goals, scene changes and framing operators. Actions, events and framing operators are STRIPS-like operators that have preconditions specifying when the operator is possible, and effects that specify how the world changes as a result of executing the operator. For instance, an action schema *SailToLand* might have the effect that the character's ship is now moored at a nearby island, and as preconditions that the character is the captain of the ship, and that the ship is not already moored somewhere. Goal schemas model what the character wants and when. Goal schemas have preconditions that determine when an agent would adopt the goal described by the schema, success-conditions that determine when the goal was achieved and failure-conditions that determine when the goal was unsuccessful. Belief schemas model some of the reasoning processes a character might go through. They are executed whenever their preconditions hold, and thus act as simple rules. For instance, a belief schema might specify that if the hatch to the hold is open, and the character has not opened it himself, then a character can adopt the belief that someone is in the hold.

An important point in the modelling of the domain is that the preconditions form the context for behaviour. This context gains coherence by not only considering the state of the story world, but also relating to past events. For instance, a pirate's goal to raise the sails can have as precondition that the captain has just ordered him to do it. The Character Agents currently use the fabula knowledge representation as a model of episodic memory to store knowledge about their life in the story world.

Modelling the domain of a particular story world is a difficult task with no clear methodology as of yet. Developing a particular story world raises questions about which actions, events and goals to add, and how to represent their preconditions and effects in terms of a knowledge representation of the story world state. The simulation is *dramatic*, meaning that we make simplifications and abstractions from the perspective of drama, rather than that of a phenomenon under study. This makes our design effort very pragmatic; we make no claims on cognitive plausibility or validity of the simulated world and its characters. A story world is a dramatic abstraction of the real world and one must make choices in level of detail, and make assessments about possible interactions, so that one can cover enough of the story world in terms of agent behaviour.

## 4.3 Character Agents

As characters in a story, the agents of the Virtual Storyteller go through a deliberation cycle in which perceptions of actions and events lead to beliefs (a type of internal element), which in turn produce emotions (another type of internal element). These emotions then lead to behaviour in terms of actions and goals for which actions can be planned.

At the core of this deliberation cycle is a partial order planner that can determine which actions to perform to reach a particular goal. It fulfils one of the basic requirements for believable characters, namely goal-directed behaviour [3]. The algorithm is similar to standard partial order planning [4, ch.11], with the following modification: if the planner cannot fulfil an open precondition for the plan using the actions it knows, it can select either an event or a framing operator. Events and framing operators are "out-of-character" actions. They are not performed by the character in the story world, so it might seem strange to include them in the character's planning process. After all, a pirate character making a plan to become rich by going to a nearby cave just so he can "accidentally" discover a hidden treasure, does not seem to be very believable. This problem occurs if the Character Agent were to execute actions that are only in the plan in order to fulfil an open precondition of a framing operator or event. Therefore, we disallow plans in which such actions occur. If an action fulfils open preconditions of a framing operator or event, but was not added to the plan for this reason (e.g., the pirate went to the cave to seek shelter from the rain), the plan is allowed.

Another important issue in the deliberation cycle is goal management. Goals can be adopted whenever their preconditions are fulfilled. At any point in the simulation, the Character Agent might have many adopted goals, and is committed to the achievement of one of them. Goals, and their outcomes, anchor

episodic structures commonly found in stories [9]. For instance, successive episodes occur when the outcome of one goal leads to the adoption of a new goal. Such episodic structures, if aimed for in the generation of stories, have implications for goal management. We use the criterium that adopted goals, beside having their preconditions met, should at least also be causally related to whatever goal(s) the agent has already adopted or acted upon. We can check this by considering the preconditions of new goals to adopt; if these preconditions are fulfilled by fabula elements that are causally related to earlier goals, we have a causal connection between this new goal and the enabling or causing fabula elements.

The Character Agents play an important role in fabula production. An implication of the aim to create fabula is that the agents must be able to explain the causality of their decisions. They need to be able to explain for instance which goal motivated their actions, which beliefs enabled them, and which perceptions caused these beliefs. We do this by propagating this responsibility over the various processes that the agent runs. For instance, the partial-order planner can explain which goal motivated the planned actions, and which beliefs fulfilled the preconditions of the action selected for execution. The system gives each element (belief, goal, action) a unique identifier, so that once the element is established in the fabula, it can also be used later for subsequent causal explanations.

#### 4.4 Plot Agent

The final responsibility for the simulation of the story world is with the Plot Agent. The Plot Agent has a facilitating role in the simulation, both outside and inside the simulated world. It exerts global control over the simulation by synchronizing its rounds. It captures the fabula elements that the Character Agents produce during the simulation and organizes them in a causal network for subsequent narration. It is also able to scale up the simulation by using *scene definitions*. A scene definition determines the setting, characters and their goals for a part of the simulation. The simplest story world simulation would consist of one scene defining the start state of the story world, and defining which characters occur in that story world. To start up a scene, the Plot Agent starts up new Character Agents if necessary, and casts them to play the required roles.

### 5 Presentation

Whereas many other systems in the field of Interactive Storytelling use 3D graphical environments to visualize the characters and the story world, in the Virtual Storyteller natural language is used as a medium to present the story to the user. The raw content for a story emerges from the simulation, as described above, in the form of a fabula. The Plot Agent passes this fabula on to the Narrator, which uses it to generate a story text. Optionally, the story can be narrated by an embodied agent using text-to-speech.

#### 5.1 From fabula to document plan

The first step carried out by the Narrator is to determine the content and the global structure of the text to be generated. This is done by converting the input fabula to a ‘document plan’: a binary branching tree containing selected elements from the fabula, connected by rhetorical relations (relations between the parts of a text). Constructing the document plan involves removing those fabula elements that will not be explicitly expressed in the story. These include positive outcomes and beliefs caused by perceptions, which are considered to be inferable by the reader. It also involves adding new elements that represent background information about the story world or properties of the characters, such as their names and locations.

When mapping the fabula to a document plan, the causal links connecting the selected fabula elements are replaced with corresponding rhetorical relations. Consecutive actions motivated by the same goal are connected using a **Temporal** relation, which can be signalled using cue phrases such as *then* and *after*. Motivation and psychological cause links are mapped to **Volitional Cause**, which implies a certain extent of intentionality on the character’s part. Cue phrases signalling this relation include *because* and *therefore*. Enablement and physical cause links are mapped to **Non-volitional Cause**, signalled by cue phrases such as *so that* and *thereby*. To introduce characters from the setting elements, the storytelling-specific **Temporal-once** relation is used, cued by *Once upon a time*. The **Elaboration** relation is used for background information, which is often expressed in a relative clause (e.g., *who was called Billy Bones*). The most general rhetorical relation is **Additive** (cue phrase: *and*); it is used if two fabula elements together cause another fabula element, and in general to connect two fabula elements if no other relation applies to them.

Next, the fabula elements in the document plan are replaced with abstract sentence structures called Dependency Trees. For each type of fabula element, a template is available specifying how its arguments should appear in the corresponding Dependency Tree. For example, the agents and patients of an action are normally given the grammatical roles of subject and object, while instruments are expressed by a prepositional argument. An example from figure 3 is *He filled the water supply with the water from a pond*. To express internal elements, there are templates for standard sentences such as *The pirate was angry* but also for storytelling-style constructions such as *He had never been so angry!*, to be used for emotions with a high intensity.

To achieve coherent output texts that are more than a sequence of simple sentences, the Narrator may combine some Dependency Trees to form complex sentences. Whether it is possible to combine two Dependency Trees depends on the cue phrase selected to express their rhetorical relation. The cue word also determines which syntactic construction is used to combine the trees. For example, in the last sentence of the example story, the cue phrase *after* introduces a subordinate clause. When Dependency Trees are combined, recurring elements may be deleted. For example, a construction such as *Billy Bones cursed and Billy Bones screamed* will be reduced to *Billy Bones cursed and screamed*.

## 5.2 Generating referring expressions

An important step in generating a fluent story is the generation of appropriate references to characters and objects. To do this, the Narrator checks (among other things) if the entity being referred to (the referent  $r$ ) has been recently mentioned, and if there are no other entities of the same gender that have been mentioned equally recently. If both conditions hold,  $r$  can be referred to using a pronoun such as *he* or *it*. However, stylistic considerations also play a role: after a pronoun has been used several times in a row, a definite description is preferred to achieve some variation. This is illustrated by the fifth sentence of the example story, which uses *a pirate* where *he* would also have been allowed.

If a regular noun phrase is used, first a noun expressing the type of  $r$  is selected (e.g., *pirate* or *ship*). Then the Narrator checks if any adjectives need to be added to the noun, for example to distinguish  $r$  from another entity of the same type. Finally, it is decided whether the noun phrase should include a definite or an indefinite article (*the* or *a*). The Narrator generally chooses an indefinite article when  $r$  is mentioned for the first time, and a definite article when  $r$  has been mentioned before. However, in some cases a definite article can be used at first mention. This is the case with so-called ‘bridging descriptions’, which refer to an entity that has already been evoked by the mention of another object it is related to. The story in figure 3 contains several examples of this, such as *the hold of his ship*, *the water supply*, and *the deck*. To be able to generate such descriptions, we have defined a number of rules stating that, for example, every entity of type ‘ship’ has a deck. If  $r$  (e.g., a deck) is related to another entity  $r'$  (e.g., a ship), the Narrator checks if there is a rule specifying that an entity of the type of  $r'$  usually has an entity of the type of  $r$ . If there is such a rule,  $r$  may be introduced using *the*. In addition, if  $r'$  has been mentioned before, and there is no other entity which may stand in the same relation to  $r$  (in the example: if there is no other ship the deck could belong to), then the relation between  $r$  and  $r'$  can be easily inferred and does not need to be mentioned explicitly. For this reason, the example story simply mentions *the deck* and not *the deck of the ship*, since the ship has already been introduced in the second sentence.

## 5.3 Text and speech

As the last text generation step, the Dependency Trees are converted into actual sentences. Using language-specific knowledge about syntax and morphology, the words in the Dependency Trees are put into the correct order, and nouns, adjectives and verbs are inflected (i.e., suffixes indicating number, tense etc. are added to the word stems in the Dependency Trees). Finally, punctuation is added.

The finished story can be presented to the user in text format, but it can also be presented by a talking face using text-to-speech. Recently, the Narrator has been coupled to DEIRA, a framework for generating spoken reports by an embodied agent [2]. The embodied storyteller can currently produce high-quality speech and lip synchronization; however the generation of appropriate facial expressions is still work in progress.



## 6 Conclusions and future work

We have described the current state of affairs of the Virtual Storyteller, a framework for the generation of stories based on simulation. We have identified particular issues with the use of autonomous characters for the simulation of stories, and described how the Virtual Storyteller addresses these issues.

Several points remain open for future work. We consider emotions and dialogue to be important aspects of storytelling, but the Character Agents are currently only capable of very simple (reactive) dialogue. Furthermore, the emotion models that we used in earlier prototypes of the Virtual Storyteller are currently under revision. We are investigating what implications story generation has for the modelling of emotion, personality and dialogue, and to what extent this needs to be made particular for a pirate domain.

We continue to explore the role that Character Agents can play as directors of their own stories. One contribution has been the use of framing operators and events to enable plans or justify the adoption of goals.

The Virtual Storyteller is currently capable of generating and presenting simple story texts. As illustrated by our example story, many improvements are still possible at the level of grammar and word choice; however, the real challenge lies in moving from straightforward reports of event sequences to real narratives, employing shifts in perspective, foreshadowing, flashbacks and other dramatic effects which go well beyond the current state-of-the-art in language generation.

Finally, the authoring process for an open-ended story generation project like the Virtual Storyteller remains a difficult issue. We intend to gain more experience in the authoring process by adding more content to our pirate domain, and are currently developing guidelines for dramatic structures that the Plot Agent can aim for by introducing a notion of conflict.

## 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] Ruth Aylett. Emergent Narrative, Social Immersion and “Storification”. In *Proceedings of the 1st International Workshop on Narrative Interaction for Learning Environments*, Edinburgh, 2000.
- [2] François Knoppel, Almer S. Tigelaar, Danny Oude Bos, Thijs Alofs, and Zsófia Ruttkay. Trackside DEIRA: A dynamic engaging intelligent reporter agent. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS’08)*, 2008.
- [3] A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.
- [4] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [5] Ivo Swartjes, Edze Kruizinga, and Mariët Theune. Let’s pretend I had a sword: Late commitment in emergent narrative. In *Proceedings of the First Joint Int. Conf. on Interactive Digital Storytelling*, 2008.
- [6] Ivo Swartjes and Joost Vromen. Emergent story generation: Lessons from improvisational theater. In *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium*, number FS-07-05 in AAAI Fall Symposium Series, 2007.
- [7] Mariët Theune, Sander Rensen, Rieks op den Akker, Dirk Heylen, and Anton Nijholt. Emotional characters for automatic plot creation. In *Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)*, 2004.
- [8] Mariët Theune, Nanda Slabbers, and Feikje Hielkema. The Narrator: NLG for digital storytelling. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG’07)*, 2007.
- [9] Tom Trabasso, Paul van den Broek, and So Young Suh. Logical necessity and transitivity of causal relations in stories. *Discourse Processes*, 12:1–25, 1989.