Hide and Sneak: Story Generation with Characters that Perceive and Assume

Hans ten Brinke, Jeroen Linssen and Mariët Theune
Human Media Interaction, University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands
j.tenbrinke@alumnus.utwente.nl, j.m.linssen@utwente.nl, m.theune@utwente.nl

Abstract

We describe the design of a perception system for the characters in the Virtual Storyteller (VST), a character-centric story generation system. Previously, these characters were omniscient; stories involving sneaking and deception could not be generated. To remedy this, we limited the characters' visual perception using simple rules. We enabled the characters to make assumptions about the story world, so they can plan toward goals in spite of incomplete knowledge. Using the distinction between the character and actor roles of agents in the VST, we can use the assumptions to steer the story plot.

1 Introduction

> *"[The little goats] were terrified and wanted to hide themselves. [..] But the wolf found them all, and used no great ceremony; one after the other he swallowed them down his throat. The youngest, who was in the clock-case, was the only one he did not find."* (Grimm and Grimm 2008)

Almost all human-authored stories, whether they are fairy tales or complex literary works, rely on the fact that characters do not know everything. Take for instance the fairy tale of the wolf and the seven little goats. Mother goat leaves her children at home and forbids them to open the door for anyone except her. However, the wolf tricks the little goats into letting him in by disguising his paws and his voice. Only one of the goats escapes by hiding inside a grandfather clock. Neither the wolf's disguise or the little goat's hiding spot would have worked if the characters had been omniscient.

There are many other examples of stories where false or missing information is key to a plot: a person is missing, an object stolen, or a character tries to reach his goals by deception. Automatic story generation systems cannot generate such stories if their characters can perceive everything that happens, anytime and anywhere; moreover, such characters are not very believable. An example is the Virtual Storyteller (VST), a system that takes a character-centric approach to story generation, where stories emerge from the actions of intelligent autonomous agents in a simulated story world (Swartjes 2010). In the original version of the VST, all character agents were provided with full and accurate knowledge of the story world. Even if something happened on the other side of the story world, all characters would know and change their plans accordingly.

In this paper we describe our approach to make the characters in the VST more believable by limiting their perceptions and allowing them to have false beliefs, without breaking the characters' ability to plan toward reaching their goals. Our solution is to allow characters to make assumptions about missing facts. These assumptions do not have to be correct: they can be used to give a character false information to steer the plot in more interesting directions. This way the planning problem can be solved, and at the same time it provides an opportunity for plot control, which is an important challenge in character-centric story generation systems (Riedl and Young 2003; 2010).

Our paper is organized as follows. First we discuss related work (Section 2) followed by a brief overview of the VST (Section 3). We then describe our main contributions: a new character perception mechanism (Section 4) and a new operator allowing characters to make assumptions about unknown facts (Section 5). We demonstrate the effectiveness of our approach with a few example stories (Section 6) and end with conclusions and future work (Section 7).

2 Related Work

Believable characters should have neither too many perceptions (the problem addressed in this paper) nor too few. The latter issue is illustrated by the classic character-centric story generation system TALE-SPIN (Meehan 1977). The concept of 'noticing' was overlooked by Meehan when writing the inference rules for his program, leading to a 'mis-spun' tale in which Bill Bird, sitting on the river bank, did not see that his friend Henry Ant fell in the river and drowned.

The Oz Project, another classic, focused on character believability (Loyall 1997) and did employ a perception system for its character agents (Bates, Loyall, and Reilly 1992). Each Oz agent runs a sense-think-act loop, selecting its next action based on its perceptions, current goals, emotional state, and other internal elements. Agents in Oz have an incomplete and possibly inaccurate view of the story world, making it possible to deceive them.

In Thespian (Si 2010), characters can also have false beliefs about the world. Each Thespian character has its own subjective world view, which includes beliefs about other characters and how they see the world ('theory of mind'). These beliefs can be revised based on characters' observations of other characters' actions. Thespian characters do not plan, but instead use a bounded lookahead policy to decide their actions. They use theory of mind to predict the short-term effects of candidate actions and select the action with the highest expected utility for their goals.

Characters in Versu (Evans and Short 2014) use a similar decision mechanism. However, instead of predicting other characters' actions, they actually execute these actions for their utility evaluation, undoing them afterwards. This way, the Versu agents avoid the risk of discrepancies between expected and actual effects of their actions.

The characters in the Othello simulation environment (Chang and Soo 2009) also have a theory of mind, and use it to make 'social plans' that include expected actions by other characters. They can use perception operators in their plans, taking into account what other characters will perceive. The system generates variations of Othello as social plans made by the Iago character. It has only been used with Iago having complete and correct information about the other characters.

FAtiMA, the agent architecture used by the emergent drama system FearNot! and its successors (Aylett, Dias, and Paiva 2006), is very similar to the VST: stories emerge from the actions of autonomous character agents that use partial order planning (POP) to decide their actions. In FAtiMA, the agents always perceive everything. However, since the stories all take place in single locations such as a class room or a school yard, this does not lead to believability problems.

Character agents in IRIS (Fendt and Young 2011) also use a POP planner. IRIS focuses on intention revision: upon failure of their plans, the characters replan or change their intentions. This is an answer to story-centric approaches such as IPOCL (Riedl and Young 2010) and CPOCL (Ware and Young 2011), where characters are given intentions to make them appear more believable, but cannot replan or change their goals. In IRIS, the characters are omniscient: they get belief updates about actions they could not realistically have perceived. Fendt and Young note that limiting the characters' perceptions would make it impossible for them to make plans that require unknown information. This is exactly the problem we address in this paper.

The problem of planning with incomplete information has been previously addressed outside the story generation domain, in particular in the context of planning for robots. Several solutions involve the use of default assumptions that are later verified through sensing (Draper, Hanks, and Weld 1994; Weser, Off, and Zhang 2010; Davis-Mendelow, Baier, and McIlraith 2012). Such work is generally aimed at maximizing the plausibility of assumptions and minimizing the costs of plan failure. However, in a story generation context, plan failure is not a big problem; it can even be a desirable feature, as it often leads to more interesting stories than plan success (Ware and Young 2011; Fendt and Young 2011). A key question therefore is how to make the assumptions believable (rather than just aiming for correctness), and how they can be used to steer the story in an interesting direction.

3 The Virtual Storyteller

The Virtual Storyteller (VST) is a multi-agent framework for story generation (Swartjes 2010). It is character-centric: the stories emerge from the actions of autonomous character agents in a simulated story world. Story generation happens in two steps: (1) plot creation and (2) language generation (Theune, Slabbers, and Hielkema 2007). In this paper, we focus on the first step.

The character agents of the VST have a dual role as actor and character. As characters, their main responsibility is to behave in a believable fashion. As actors, the agents bear a responsibility for creating an interesting plot. The character agents go through a deliberation cycle in which perceptions of actions and events lead to beliefs. These beliefs then enable new actions or lead to the adoption of goals for which actions can be planned. When a character performs an action, the story world is updated and all agents are informed of the change: they 'perceive' it.

At the core of the character agents' deliberation cycle is a POP planner that can determine which actions to perform to reach a particular goal. The planner tries to make a complete plan from start to finish, which will fail if crucial knowledge is missing. The VST uses a closed world assumption, meaning that any fact not explicitly known to be true is assumed to be false. If the location of an object is unknown, the planner concludes that making a plan to get that object is impossible.

Multiple story domains have been defined for the VST, including a classical fairy tale domain with knights and princesses, a story world inspired by Little Red Riding Hood, and a pirate domain. For the research presented in this paper, we made a variation of the pirate world. The setting is a pirate ship with various locations; the main characters are O'Malley, the ship's cat, and Scurvy, the bilge rat. Scurvy wants to eat the cheese that is located somewhere on the ship and O'Malley has the goal to catch Scurvy.

4 A New Perception System

To keep our characters from being omniscient, we have created a new perception system based on the distinction between character (*in-character*, IC) and actor (*out-of-character*, OOC) knowledge: characters make plans based on IC knowledge, whereas the actors' OOC reasoning is done based on the full and accurate knowledge they require.

In this section, we describe the perception rules characters use to perceive their world and how their IC knowledge is updated. We also address how the characters' initial knowledge of the world has to be set up. We only deal with visual perception; other senses are left for future work.

4.1 Visibility Rules

The rules for visibility are as follows. Characters, objects and actions at the same location as the character can be seen by the character; this is similar to the perception rules of Chang and Soo (2009). The character has a view of other locations if there is a direct path to that location that is not blocked by a closed door. Objects in a container can be seen if it is open and located at the character's location.

For perception of an object, the rules work in three steps. Step (1) checks whether an object is at a location. Step (2) determines from which locations the object is visible, and step (3) checks if an object is visible for a character, using the previous two steps.

Visibility rules are an important part of the description of the story world and, while these basic rules are domain-independent, authors can easily expand them for domain-specific purposes; the responsibility of keeping them in line with the story world is left to the author. This requires more authoring effort, but also allows for more flexibility. If, for example, authors want to add magical crystals that allow a character to see distant worlds, they can. It would not be possible to simply capture such creative solutions in a general system that works for all story worlds.

4.2 Dealing with Perceptions

The next step is to apply the visibility rules to incoming perceptions. We take our inspiration from Gordon and van Lent (2002), who discuss the use of virtual humans as participants (characters, IC) versus virtual humans as actors (OOC). The choice between the two affects how agents perceive the world: virtual participants should only perceive those parts of the world they can realistically perceive, while actors should be omniscient. However, actors should still be aware of the realistic bounds of their perceptions to remain believable.

The first thing a VST character agent should do with a new perception is handle it OOC so the actor's knowledge stays up to date. The perception is then analyzed using the visibility rules to see whether the character gets to see it IC. If this test returns that the object or character is not visible, the perception should not be added as a belief to the character's IC knowledge base. Because the characters only get updates (new perceptions) when changes occur in the world, we do not discard the facts that are not perceived IC. Instead, we store them OOC, so that they can be added if they become available to the character at a later time. Thus, whenever a new perception is received, previously missed facts are re-tested for visibility. This amounts to taking care of two different cases. The first case concerns beliefs that must be added due to the effects of the new perception. For instance, if O'Malley sees that Scurvy is at the deck, this means he will conclude, based on OOC knowledge and the axiom that people can only be in one place at a time, that Scurvy is no longer in the galley. The second case involves facts that have to be removed because they are no longer true. For example, Scurvy believes IC that the cheese is in the galley. When he arrives there his IC knowledge suggests that he should be able to see the cheese; however his OOC knowledge reveals that he cannot. The OOC knowledge is always correct, therefore the belief has to be false and is dropped.

4.3 Initial Character Knowledge

In the VST, characters get their initial knowledge of the world at the start of a run of story generation. Without this, they have an empty IC knowledge base and will be unable to plan. We give each character a separate copy of the state of the world, but allow this to contain false information. By changing a few things in the characters' initial knowledge, their behavior can be manipulated to suit the story. For example, we can make sure O'Malley does not know IC where Scurvy is at the start of the story, so that he cannot plan to catch the rat until he first sees him. It also allows us to give Scurvy a false belief about the location of the cheese.

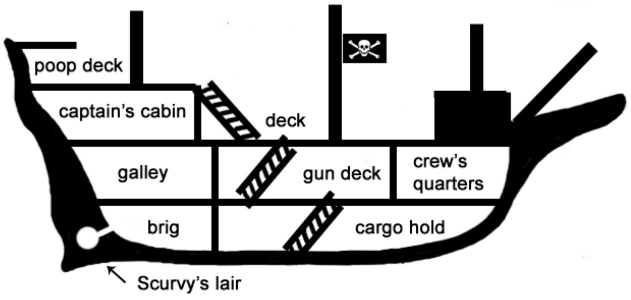

Figure 1: Schematic map of the ship in the storyworld.

To test the visibility rules, we created a scenario in which Scurvy has a goal to get the cheese and O'Malley wishes to catch Scurvy. Figure 1 shows the schematic layout of a pirate ship which is the basis for the story world. We gave Scurvy the initial belief that the cheese was in the captain's cabin, leading him to go to this location. After opening the door to the captain's cabin, he noticed the cheese was not there and the story ended. As expected, Scurvy did not know where else to find the cheese and could not make a plan to get it. Without any other goals to pursue, he did nothing and got caught by O'Malley. In the next section, we discuss how we let characters use assumptions about the story world to overcome this planning problem.

5 Planning with Assumptions

As we saw in the previous section, limiting the characters to using only IC beliefs for planning introduces a new problem. If a character has the goal to acquire some object, it is unable to make a plan to do so if it does not know where the object is and it will drop the goal. This of course does not lead to interesting stories. What we need is a way for characters to fill in the missing information.

Multiple approaches could be used to give the characters more information. For example, social interactions could be used to share information between characters, as in Thespian or Versu (Si 2010; Evans and Short 2014). However, implementing character dialog was beyond the scope of this research. What we want is for the characters to be able to make plans based on incomplete knowledge, without making drastic changes to the planner or the rest of the system.

5.1 Assumption operators

Since characters can now have incomplete or inaccurate knowledge of the world, the closed world assumption no longer holds IC. However, it still holds for the OOC part of the agents. When a character does not know the location of an object, instead of believing that the object is not located anywhere, it should assume that the object is somewhere that it cannot currently see. To make this possible, we implemented an assumption operator that works similarly to other planning operators.

Having separate assumptions for every object that a character can lose track of would soon become unmanageable for larger story worlds, so a general operator is needed. The schema for a general `AssumeObjectAtLocation` assumption has three arguments: the Character making the assumption, the Object the assumption is about, and the Location where the object is assumed to be. The effect of the assumption operator is a fact that puts the Object at the Location, but only in the Character's mind (IC).

The assumption operator does not need to provide the correct location for the object. If it did, the situation would not be much better than having omniscient characters. It should be possible for a character to search in the wrong location. The planner is not looking for an optimal solution like other approaches (Draper, Hanks, and Weld 1994; Davis-Mendelow, Baier, and McIlraith 2012); it is no problem if false information is used to make a plan. At some point during plan execution, a perception will trigger and the false information will be removed from the knowledge base. At this point the plan will become invalid and a new plan will be made if possible.

Some constraints on assumptions are needed; otherwise the planner would simply assume the object to be at the character's location, as this would result in the shortest plan. What's more, the object would be assumed to be at a more convenient location even if its actual location is known but going there would result in a longer plan.

Clearly, if a character knows the location of an object, it should not make any assumptions that contradict this knowledge. This also means the object cannot be at the character's current location; if it was, the character would have seen it. In addition, the object must exist: the actor must have OOC knowledge about the location of the object. Otherwise, the character might begin a search that will never end.

To make sure the characters search in a believable manner we take our inspiration from the Oz system, where agents use current sensory information as well as past knowledge to select possible plans (Bates et al. 1992). The characters in the VST also keep track of past events, which we can use to find out where a character has been and how long ago this was. This way, we can exclude a location as a believable search location if the character has been there too recently.

See Figure 2 for two rules that capture these general constraints on search locations.

5.2 Executing assumptions

Assumptions are implemented as proper operators that have to be executed. Unlike other operators they do not affect the actual world; they only change the internal world representation of the character that makes them, so their effects are IC only. The character stores the effects of the operator directly as beliefs. Whenever a character executes an assumption operator there is a chance that the resulting belief does not correspond to the actual world state. However, the mechanism described in Section 4.2 allows the characters to deal with such false beliefs.

properSearchLocation(?loc, ?obj)	
at(?obj, ?loc3)[OOC] ∧	// *True object location known OOC*
¬(at(?obj, ?loc2)[IC] ∧ ?loc2 ≠ ?loc)	// *No IC belief about*
	// *object at different location*

believableSearchLocation(?loc, ?char)	
visitTime(?char, ?loc) = NIL ∨	//*Location never visited, or*
currentTime – visitTime(?char, ?loc) > 10	//*over 10 turns ago*

Figure 2: General constraints on search locations.

AssumeObjectAtEnemyLocation(?char, ?obj, ?loc, ?char2)	
Preconditions	
enemy(?char, ?char2) ∧	// *Characters are enemies*
at(?char2, ?loc)[OOC] ∧	// *Enemy location known OOC*
properSearchLocation(?loc, ?obj) ∧	// *Location meets general*
believableSearchLocation(?loc, ?char) ∧	// *constraints*
¬at(?char2, ?loc)[IC]	// *Enemy location not known IC*
Effects	
at(?obj, ?loc)[IC]	// *Assume object at enemy location*

Figure 3: Assumption operator placing an object at the location of another character (an enemy).

With these changes, the character agent properly executes assumptions and also retracts them if they prove to be false. Instead of giving up when losing track of an object's location, the character will search for the object methodically, starting with the nearest possible location within the constraints from Figure 2, because this will provide the shortest plan. Eventually the character will find the desired object and continue with the rest of his plan. Whether or not the resulting story is interesting is left to chance. During its search, the character might encounter other things it was unaware of. It may run into conflicts it was not expecting. But then again, it may also get lucky and guess right on the first try, avoiding any trouble.

In small story domains we can manipulate the initial location of characters and objects to ensure that something interesting will happen. But in bigger and more complex scenarios, controlling the location of objects and especially characters that way will not be possible. To circumvent this problem, we can take assumptions one step further.

5.3 Plot control

When a character makes an assumption, this is an ideal moment to exert control over the plot without impacting the character's believability. The priority for the actors in the VST is not to reach their character's goal, but to reach the *meta-goal* of creating an interesting story. A story world may have opportunities for events that, while interesting for the audience, are not in line with any of the characters' goals. If the agent has OOC knowledge about these opportunities, it can direct the character towards them, instead of directly towards the character's goal. At a moment when a character no longer knows where to go, the actor can steer the character towards conflict without limiting the character's autonomy.

Ideally we would want to give the actors explicit meta-goals to help make the story more interesting, but for now we use a simpler, more direct approach to give an impression of how this could work. We make separate assumption schemas for each phase of the story, with additional preconditions to either introduce conflict or resolve it. This is similar to making actions available to a character depending on the story phase, as proposed by Tommassen (2009): i.e., before or after the climax of the story. For the first half of the story we can make an assumption that guides the character towards a conflict. Figure 3 shows an example of an assumption schema that could be used to let a character assume an object is at a location where its enemy just 'happens' to be (known OOC only). After the main conflict has been resolved we can use another assumption (a correct one this time) to tie up loose ends and finish the story.

It is the responsibility of the author to set the assumptions in such a way that interesting stories and believable behaviors will emerge. This can be done using an *iterative authoring* approach (Swartjes and Theune 2009), by manually authoring the schemas and story world and then iteratively generating stories and editing the authored content based on the results, thus co-creating the stories with the system.

6 Evaluation

In this section, we show how the characters' new perception and assumption capabilities can be used to author interesting stories. Like before, we use a story world in which Scurvy wants to obtain the cheese. We do not tell him where the cheese is, so he will have to search for it. To make the stories more exciting, we make sure that Scurvy runs into O'Malley, who will try to catch Scurvy as soon as he sees him.

6.1 Expanding the Story World

We use an expanded version of the story world from Section 4.3. Previously, Scurvy only had the goal to get the cheese and therefore he would just ignore O'Malley's presence. To prevent this, we add an `escape` goal that will make Scurvy flee once he sees O'Malley. Once the escape goal has been triggered, Scurvy will try to run from O'Malley. Since they both move at the same speed, Scurvy will either get backed into a corner or he will find the location of the cheese. In both cases, Scurvy will need a way to keep O'Malley busy, otherwise he will not have enough time to get away or pick up the cheese. For this purpose we add a `distract` goal, which can be achieved by some new actions reminiscent of the Tom & Jerry cartoons. The rat can drop a cannonball on O'Malley's foot, preventing him from following, and he can kick over a bucket of soapy water, which will make the deck slippery. For O'Malley there are two new actions that he performs in response to Scurvy's distracting actions: dancing around on one foot while screaming in pain, and cleaning the deck. We also give O'Malley the goal to swab a deck if it is slippery, to keep the captain from getting angry with him because of an untidy ship. As a consequence, if Scurvy makes the deck slippery, O'Malley is forced to swab the deck. Finally, Scurvy gets a new action that lets him use the tunnel to get to his lair.

To make the story progress the way we want it to (as discussed in Section 5.3), we add two assumption operators. The first makes Scurvy assume the cheese is at the location where O'Malley is, by using the actor's OOC knowledge (see Figure 3). This can be used early in the story to ensure that Scurvy will run into O'Malley and begin the conflict. The second assumption lets Scurvy assume the cheese is at the correct location. This can be used at any point after O'Malley has been distracted at least once and thus the conflict has already taken place. This guarantees that, from then on, the story will start to work towards a resolution.

6.2 Example Stories

In our first story, we let Scurvy start at his lair and O'Malley at the galley. The cheese is placed in the captain's cabin; there is a bucket with soapy water on the deck and a cannonball on the gun deck. Scurvy immediately takes on the goal to find a piece of cheese. Figure 4 shows a transcript of the story as it is played out. Lines describing characters' perceptions and assumptions are shown in boldface. The story is not very fluent, as we did not use the language generation component of the VST for narration.

Scurvy wants to get the cheese.
Scurvy assumes the cheese is in the galley.
Scurvy walks to the cargo hold via the door.
Scurvy walks to the gun deck via the ladder.
Scurvy opens the door to the galley.
Scurvy can see the cheese is not in the galley.
O'Malley sees Scurvy.
O'Malley wants to catch Scurvy.
O'Malley walks to the gun deck via the door.
Scurvy wants to avoid O'Malley.
Scurvy walks to the crew's quarters via the door.
Scurvy wants to distract O'Malley.
Scurvy walks to the gun deck via the door.
O'Malley walks to the crew's quarters via the door.
O'Malley walks to the gun deck via the door.
Scurvy picks up the cannonball from the gun deck.
Scurvy drops the cannonball on O'Malley's foot.
Scurvy assumes the cheese is in the captain's cabin.
Scurvy walks to the deck via the ladder.
O'Malley dances around on one foot, screaming loudly.
Scurvy opens the door to the captain's cabin.
Scurvy walks to the captain's cabin via the door.
Scurvy picks up the cheese from the captain's cabin.
O'Malley walks to the deck via the ladder.
Scurvy wants to hide with the cheese at Scurvy's lair.
Scurvy walks to the deck via the door.
O'Malley walks to the captain's cabin via the door.
O'Malley walks to the deck via the door.
Scurvy walks to the gun deck via the ladder.
Scurvy walks to the cargo hold via the ladder.
O'Malley walks to the gun deck via the ladder.
Scurvy walks to the brig via the door.
O'Malley walks to the cargo hold via the ladder.
Scurvy crawls to Scurvy's lair via the small hole.
O'Malley walks to the brig via the door.

Figure 4: First example story generated by the VST.

Since Scurvy does not know where the cheese is, he has to make an assumption about its location. As the story has just started and no conflict has taken place yet, the agent uses its OOC knowledge about O'Malley's location (the galley) to assume that the cheese is at that location (2). As soon as Scurvy opens the door, O'Malley discovers him and starts to chase him (7-9). Scurvy tries to run away, but finds himself cornered in the crew's quarters. He then decides he has to distract O'Malley to get away and uses the cannonball to achieve this (13, 16). As O'Malley is now distracted, Scurvy can get back to the goal of finding the cheese. He has found out that the cheese is not at the galley, so he needs to make a new assumption. Since the conflict has now taken place, a new story phase is entered in which the agent's OOC knowledge is used to guide Scurvy towards the cheese (18). This helps to maintain the pace of the story instead of dragging it on with uninteresting searches in wrong locations. Scurvy now goes straight to the cheese's actual location, the captain's cabin (22). He grabs the cheese and adopts his final goal of taking it back to his lair (25). In the meantime, O'Malley has recovered from his sore foot, but as Scurvy is constantly moving, O'Malley is unable to catch him before Scurvy makes it safely back to his lair (34). There O'Malley can no longer reach him, so the story ends.

For our second example story we change the starting locations of the characters: Scurvy begins on the poop deck and O'Malley begins on the gun deck. The cheese is still in the captain's cabin and, again, Scurvy starts off by making a false assumption leading him to O'Malley. O'Malley chases Scurvy who is forced to retreat to the poop deck. Unable to move further in this direction, Scurvy spots a bucket of soapy water on the deck and distracts O'Malley by kicking it over. O'Malley is obligated to clean the deck, giving Scurvy a chance to go for the cheese. Again, Scurvy uses OOC knowledge to assume where the cheese is, because the conflict has already taken place. O'Malley finishes his task just as Scurvy is on his way back from the captain's cabin with the cheese. He chases Scurvy again, but, as before, the rat escapes through his tiny tunnel out of O'Malley's reach.

In the second story, the bucket of soapy water was located more conveniently than the cannonball, so Scurvy used that in his plan to distract the cat. Other than that, the dramatic arc was similar to the first story: the hero sets out on an adventure, has to search for his prize, runs into the villain, is chased into a corner, defeats the villain, finds his prize and finally escapes with the villain breathing down his neck.

The generated stories illustrate how the split between IC and OOC knowledge can be used. The characters always try to make use of their IC knowledge and base their plans on that as long as the information is sufficient to make a complete plan. Whenever beliefs are missing or prove to be false, OOC knowledge is used in combination with past beliefs from the episodic memory. The latter helps the characters to make believable choices as they will not search in places they have recently visited. The former is used to direct the characters towards interesting places depending on the current state of the plot (see Section 5.3). The stories also show that characters can be deceived, as in fact the actor component deceives the character and gives it false beliefs.

7 Conclusions

We have shown how we turned the omniscient characters in the Virtual Storyteller into more believable ones whose knowledge of the story world is limited by what they can realistically perceive. To this end, we implemented a set of perception rules that limit the characters' knowledge about the story world. To compensate for this loss of information, we implemented an assumption operator that allows characters to make a believable guess about the missing facts and use it in planning. If the assumption is wrong, the character simply tries again until it gets it right.

Since random searching does not add much excitement to the story, we implemented plot control options through the assumption operators. When the character has to decide where to go, we have a good opportunity to direct the character towards a potentially interesting situation. The character agents in the VST have separate knowledge bases for in-character (IC) and out-of-character (OOC) knowledge. The first stores the character's (possibly incorrect) beliefs about the story world and the second stores the actual world state for the actor. Making use of OOC knowledge, the character can be directed either to the correct location or to a location that adds some conflict and excitement to the story.

We generated a number of stories showing that the perception system and the assumption operators work as intended. The characters now act on beliefs that they could plausibly have instead of being aware of every change in the world. The guided assumptions let the characters make wrong decisions and can be used to force conflicts to happen.

The assumption operators used in the example stories were tailored specifically to our story domain. The only interesting locations were the location of O'Malley and the actual location of the cheese. In the future more options should be added. For this, we would like to use a more high level approach. By giving the actors meta-goals to achieve, they can decide which are the most interesting assumptions. For example, if the meta-goal is to start a conflict, the actor could pick from a number of available conflict descriptions and try to fill in the assumption in such a way that the character's resulting plan will achieve the preconditions that will cause the conflict. Actors would also have to keep track of the goals of other characters so they can anticipate their moves, i.e., they would need a theory of mind.

So far we have only implemented the sense of sight. Adding the other classic senses (hearing, touch, smell, taste) would be good to increase the number of stories that can be told. Hearing in particular will become more important when the characters are given the ability to speak and exchange information; e.g., allowing stories that involve eavesdropping.

Finally, we need to test our general approach with different and more complex story domains, and carry out user experiments to evaluate character believability and interestingness of the generated stories.

Acknowledgments

This publication was supported by the Dutch national program COMMIT. We thank Ivo Swartjes for his advice in the initial stages of this research.

References

Aylett, R.; Dias, J.; and Paiva, A. 2006. An affectively driven planner for synthetic characters. In *International Conference on Automated Planning and Scheduling*, 2–10.

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1992. An architecture for action, emotion, and social behavior. In *Selected papers from the 4th European Workshop on on Modelling Autonomous Agents in a Multi-Agent World, Artificial Social Systems*, 55–68.

Chang, H.-M., and Soo, V.-W. 2009. Planning-based narrative generation in simulated game universes. *IEEE Transactions on Computational Intelligence and AI in Games* 1(3):200–213.

Davis-Mendelow, S.; Baier, J. A.; and McIlraith, S. A. 2012. Making reasonable assumptions to plan with incomplete information: Abridged report. In *Problem Solving Using Classical Planners, AAAI Technical Report WS-12-12*, 16–22.

Draper, D.; Hanks, S.; and Weld, D. S. 1994. Probabilistic planning with information gathering and contingent execution. In *Proceedings of the Second International Conference on AI Planning Systems (AIPS)*, 31–36.

Evans, R., and Short, E. 2014. Versu – A simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2):113–130.

Fendt, M. W., and Young, R. M. 2011. The case for intention revision in stories and its incorporation into IRIS, a story-based planning system. In *Proceedings of the Intelligent Narrative Technologies Workshop*, 10–16.

Gordon, A., and van Lent, M. 2002. Virtual humans as participants vs. virtual humans as actors. In *Proceedings of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*.

Grimm, J., and Grimm, W. 2008. The wolf and the seven little kids. In *Fairy Tales*. Salt Lake City, UT: Project Gutenberg.

Loyall, B. 1997. *Believable Agents: Building Interactive Personalities*. Ph.D. Dissertation, Carnegie Mellon University.

Meehan, J. R. 1977. TALE-SPIN, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91–98.

Riedl, M. O., and Young, R. M. 2003. Character-focused narrative generation for execution in virtual worlds. In *Proceedings of the International Conference on Virtual Storytelling*, 47–56.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39:217–268.

Si, M. 2010. *Thespian: A Decision-Theoretic Framework for Interactive Narratives*. Ph.D. Dissertation, University of Southern California.

Swartjes, I., and Theune, M. 2009. Iterative authoring using story generation feedback: Debugging or co-creation? In *Proceedings of the Second Joint International Conference on Interactive Digital Storytelling*, 62–73.

Swartjes, I. 2010. *Whose story is it anyway? How improv informs agency and authorship of emergent narrative*. Ph.D. Dissertation, University of Twente.

Theune, M.; Slabbers, N.; and Hielkema, F. 2007. The automatic generation of narratives. In Dirix, P.; Schuurman, I.; Vandeghinste, V.; and van Eynde, F., eds., *Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands*, 131–146.

Tommassen, P. 2009. Wacky pirate hijinks – Causing and resolving conflict between autonomous agents in a virtual storyworld. Master's thesis, University of Twente.

Ware, S. G., and Young, R. M. 2011. CPOCL: A narrative planner supporting conflict. In *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 97–102.

Weser, M.; Off, D.; and Zhang, J. 2010. HTN robot planning in partially observable dynamic environments. In *International Conference on Robotics and Automation*, 1505–1510.