Late Commitment
Virtual Story Characters that can Frame their World

Ivo Swartjes and Mariët Theune

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

Abstract. Our long-term aim is to build virtual agents that can partake together with human interactors as characters in a story, which emerges from their interactions wich each other and with the story world (the emergent narrative approach). Improvisational theatre – as a real-life example – suggests an extension of the current approach to emergent narrative: the event sequence of a story does not emerge based on a fixed story world setting, but the event sequence and the story world setting underlying it, emerge together. We propose a model of this idea, which we call *late commitment*, meaning that choices about aspects of the story world setting are delayed until they become useful for story progression. We discuss possibilities and issues concerning the use of this technique in the context of action planning and goal adoption for character agents.

1 Introduction

When building virtual characters whose interactions should produce engaging drama, there are two concerns at play. First, we would like them to display interesting aspects of their personalities, so we can find out what drives them and affects them. Second, we would like to see these aspects displayed within a plot, giving a sense of dramatic development, coherence and closure to the event sequence. One does not necessarily follow from the other, as has also been noted in the interactive storytelling community, where it has led to research into the question how to sensibly and effectively integrate character and plot for the dynamic creation of stories that a human user can interact with (e.g., [1, 2]).

An attractive separation of concerns has been to implement the notion of ‘character’ by means of an autonomous agent, whose personality, goals, emotions and actions can be modeled using state-of-the-art research in fields such as cognitive modeling and AI. The plot might then be more or less guided by a drama manager component that gives directions to these characters so they ‘do the right thing’ for a desired plot. However, directing autonomous characters for the purpose of interactive storytelling is fundamentally problematic, as argued convincingly by Mateas [1]: the drama manager’s guidance cannot be loosely coupled to an autonomous character unless the character offers specific ‘hooks’ for guidance, in which case it might not be useful to make the characters autonomous in the first place. An alternative is that the plot might emerge directly from the resulting character interactions: the *emergent narrative* approach [3].

However, a successful emergence of plot is not guaranteed in this approach, as there is no explicit support to make the event sequence achieve a coherent and finished whole.

There is a similarity between emergent narrative and improvisational theatre [4]. Improvisational actors are also not directed: story construction is an emergent process, whereby each actor shares responsibility for the story development, but none of the actors can control its outcome. For virtual agents, this suggests a *distributed drama management* approach [5], where each agent has the means to reason about, and contribute to, the emerging story. In this approach, we argue for an explicit consideration of two roles for the design of a virtual character agent: the role of character *in* a story (its in-character or IC role), as well as the role of actor *of* that story (its out-of-character, or OOC role). See Fig. 1.

Modeling the IC role can draw heavily from AI research on virtual characters. For instance, the emergent narrative demonstrator FearNot! [3] is based on state-of-the-art research in emotion and cognitive agent models. FearNot! deals with the drama of bullying in preschool classrooms. The agent architecture underlying the characters in FearNot! (called FAtiMA) uses a cognitive appraisal model to determine the best course of action for a character. A child that is being pushed, might for instance become distressed, and decide to push back, or start crying.

The OOC role is not a separate layer, loosely coupled to the IC role, because this would introduce the same kind of problems that drama manager directives would introduce. Rather, it is a tightly coupled extension of the IC role with ways to control, constrain and adapt IC processes for the sake of interesting story development. The assumption of this approach is that it is possible to determine heuristics for story development that do not rely on (1) planning for other characters or (2) planning too far ahead. One such heuristic is the consideration of the emotional impact of selected actions: if actions impact other characters emotionally, they might benefit the dramatic value of the story. This heuristic was investigated using an OOC mechanism called *double appraisal* [6], an extension of the FAtiMA agent architecture to predict the emotional reaction by other characters to possible actions of the agent. Taking this prediction into account when choosing actions for execution has increased the dramatic value of produced stories in comparison to stories produced without this mechanism.

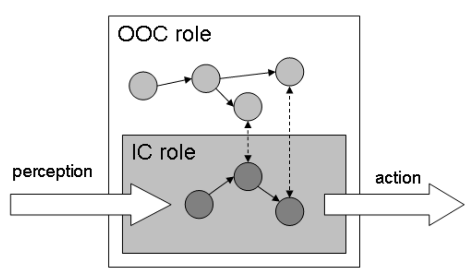

Fig. 1. Two tightly coupled roles of a character agent: in-character (IC) and out-of-character (OOC). The circles represent cognitive processes.

The context of this research is the development of the Virtual Storyteller, a story generator based on the emergent narrative concept, where virtual character agents 'live their lifes' IC, interacting with each other and the virtual world, leading to an event sequence that is subsequently re-told in natural language and optionally presented using speech by an embodied agent [7]. The goal of the Virtual Storyteller is not so much to support user participation, but to investigate under which conditions the simulation of virtual characters can lead to the emergence of interesting stories.

To further define the OOC role of the character agents in the Virtual Storyteller, we have started studying techniques that are used by improv actors to accomplish the process of story co-creation in improvisational theatre [4]. We focus in this paper on the technique of retrospectively establishing aspects of the story world in the course of the story creation process. We have implemented a model of the technique called *late commitment* [8, 9] and make a first exploration of the potential for virtual characters to fill in the setting of the story world in the course of the simulation, making it part of the emergent process rather than fully specifying it beforehand[1].

The rest of the paper is organized as follows. Section 2 describes the idea of late commitment in more detail and argues why it may be benificial to emergent narrative. Section 3 introduces the notion of *framing operator* as a way to model late commitment choices. Section 4 describes how and why we are currently using framing operators in the Virtual Storyteller, followed by a discussion of some of the consistency issues that arise when framing operators are used in Sect. 5. We conclude and describe future work in Sect. 6.

2 Late Commitment

We make a distinction between (1) the *setting* of the story world: the characters, their personalities, relationships, backstories, objects and properties of objects that exist at the point in time where the story starts, and (2) the *event sequence* of the story, emerging forward in time, i.e., the actions that characters perform, the thoughts and emotions they experience and the goals they adopt, simulated based on a particular setting.

The current method of emergent narrative is to fully define the setting at authoring time, and then start a simulation, leading to a particular event sequence at run time. There are two problems with this method, both having to do with the big impact that the setting has on the event sequences that result from it. First, a fixed setting constrains the possible courses of events considerably, taking away part of its generative potential. Moreover, the relationship

[1] Late commitment should not be confused with the term *least commitment*, commonly used in the planning community. While both have a similar rationale (i.e., keeping options open by delaying choices as long as possible), least commitment refers to deferring decisions about variable bindings and step orderings in the search for a plan, whereas late commitment refers to deferring decisions about the story world setting in the course of the simulation.

between setting and event sequence is unclear at authoring time, making it difficult to make informed choices about the setting such that certain desired event sequences can occur.

In contrast, in improvisational theatre the event sequence is not a result of a given setting, but the definition of the setting is retroactive and emerges alongside the event sequence, through offers given by the actors [10]. This process can be illustrated with the following short example scene, which could have been improvised by three actors (A, B and C). Examples of knowledge about the setting emerging during the dialogue are included between parentheses (alternative interpretations are very well possible):

A: Still up? *(B is awake, it is late)*
B: Yes dear, I couldn't sleep. I have an important meeting tomorrow with my boss. I think he found out that I sometimes fall asleep at work. *(A and B are husband and wife,there is a boss, backstory: appointment made, subject sometimes falls asleep)*
A: Yea, admittedly not the best quality for a security guard. *(B works as security guard)*
B: [*out-of-character*] The next day, at work...
B: [*to* C] Hi, boss. [*big yawn*] *(B is tired, C is the boss)*
C: Well well, Mr. Peters, you look like you are about to fall asleep at work...again! *(B is called Mr. Peters, backstory: C has indeed seen B sleep)*
. . .

The technique of *late commitment* described in this paper models this process of gradually establishing the setting of the virtual story world rather than having it fixed from the start of a simulation. Clearly, this is an OOC process. Development of the technique has two aspects: (1) defining a technical solution for giving the characters the OOC ability to define aspects of the story world, in such a way that they can pretend IC that these aspects have been the case since the start of the simulation (described in Sect. 3), and (2) determining when and why certain aspects of the story world might be added at a given point in the simulation (described in Sect. 4).

3 Modeling Late Commitment

Using late commitment requires a mechanism to determine the conditions for asserting aspects of the setting in retrospect without introducing inconsistencies with what was already defined. For story planning, Riedl and Young have explored this issue with their Initial State Revision (ISR) algorithm [11]. ISR planning is a partial order planning algorithm that allows one to specify the initial state of the planner (representing the setting of a story to be planned) in terms of facts that have indeterminate truth value. The planner can make choices to make these facts true or false in order to resolve open preconditions of a partial plan. To avoid inconsistencies (e.g., an object cannot be located at two different places), such indeterminate facts are organized into *mutex sets*. A mutex set is a set of facts with undetermined truth value, of which only one can become true. Once one of the facts in the set becomes true, the other ones automatically become false.

To enable late commitment, we adopt an approach similar to the approach of Riedl and Young, but model incomplete or undefined world states in a way that is more expressive. We add a new type of operators to the story domain, which we call *framing operators*. Just like other operators used in the simulation (i.e., operators that represent character actions and story world events), framing operators have preconditions and effects. The difference is that the effects of a framing operator specify a *commitment* (rather than a change) to facts of the story world setting. Normal actions are performed IC (i.e., relate to the agent's character) whereas framing operators are OOC (as if the actor says to the rest of the agents: "Let's pretend my character has always hated your character."). The execution of a framing operator should create the IC illusion that it did not occur, and that its effects have been true since the start of the simulation. The preconditions of the framing operator can be used to avoid inconsistencies (just like mutex sets), but also provide a mechanism to define a proper contextualization for the operator. An example is the `CarryRapier` framing operator in Fig. 2, which commits to a fact in the setting that specifies that a certain character is carrying a rapier. The preconditions contextualize the operator in an author-defined way, answering the following question: when is it believable that a certain character happens to be carrying a rapier? The `CarryRapier` operator specifies that this is the case when the character is a pirate, and the rapier is owned by this pirate. Furthermore, the rapier cannot be at two locations at the same time, so this inconsistent situation is also excluded by the preconditions.

CarryRapier	
Preconditions:	*Effects:*
`pirate(?char)`	`at(?rapier, ?char)`
`rapier(?rapier)`	
`owns(?char, ?rapier)`	
¬ `at(?rapier, ?loc)`	

Fig. 2. The `CarryRapier` framing operator. Whenever there is a pirate that owns a rapier, and the rapier is not (yet) located anywhere, the `CarryRapier` framing operator can be used to commit to the fact that the pirate is carrying the rapier.

The choice which framing operators to include in the story domain, and what the preconditions are for these operators, is up to the author of the story domain. For example, the author can choose to add a `HateYou` framing operator, that establishes that character A hates character B, with as only preconditions that A and B do not have a friendly relationship. This would make for somewhat flat characters that can just happen to hate each other if they so choose (OOC).

The characters can be made rounder by adding a precondition to `HateYou` that requires a reason for the hate. For instance, `HateYou` can have a backstory precondition that the hateful character A was offended by B. Besides adding ways for characters to offend each other, one might add framing operators that establish offences, for instance a `BeenCalledCoward` operator, which commits to the backstory fact that A had been called a coward by B (in narrative time before the simulation), and that he was offended by that incident. This way, framing operators can build on each other to further establish the setting of the emerging story at run time.

4 Using Late Commitment in the Virtual Storyteller

In the Virtual Storyteller, we currently use late commitment for two purposes. The first is to allow the character agents to justify the adoption of new goals; the second is to support their action selection process. Both processes make the elements introduced through late commitment narratively necessary to a certain extent.

For the normal IC process of goal adoption in the Virtual Storyteller, character agents check whether the preconditions of author-defined goals are met. For instance, adopting a goal of plundering another ship might require that there is another ship in sight, and that the character adopting the goal is a pirate captain. If these circumstances do not apply, the Virtual Storyteller can decide one of two things: (1) not adopt the goal or (2) *justify* the goal (OOC) by adapting the circumstances using late commitment. For instance, a ship in sight might be framed, and the character can be endowed with the role of pirate captain, making the preconditions valid. A first naive heuristic determining when to justify a goal is that the character agents try to always have goals to pursue. As soon as they do not have any more goals that they can pursue or adopt, the character agents will search for new goals that they can justify by establishing the preconditions through late commitment. A next step might be that the agents attempt to justify and adopt goals that conflict with important goals of other characters, to facilitate the emergence of interesting dramatic conflict.

The character agents also use late commitment to support action selection. They use a partial order planner to select actions that achieve their goals. Framing operators support this planning process by adapting the circumstances to make plans possible. For instance, if an agent that plays the role of captain of a ship has adopted a goal to find out whether an approaching ship is friend or foe, a possible plan might involve the use of binoculars to get a closer look. If these binoculars do not exist yet, the planner might not find a plan, but if it can use a framing operator that commits to them being in the captain's cabin, it can create a plan involving the captain going to his cabin to get them.

Both goal justification and action selection make use of a partial order planner that we adapted to allow interweaving of selecting IC operators such as character actions, and OOC operators such as framing operators and story world events. We first give an example to illustrate the process of partial order planning with

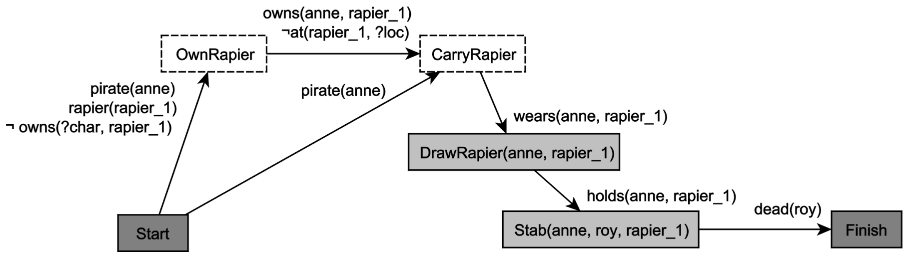

Fig. 3. Partial order plan using framing operators. The arrows represent causal links of the plan; the dashed rectangles represent framing operators.

framing operators, and then proceed to discuss the modifications made to the planner to support it.

4.1 Example

We first define a small story domain without using late commitment. We include two pirates in the setting: Anne Bonney and Rapier Roy. We define that Anne Bonney hates Rapier Roy, and endow her with a rapier. These seemingly arbitrary choices have a big impact on the simulation, where Anne Bonney's hatred towards Rapier Roy makes her adopt a goal to kill him. This leads to the construction of a plan to draw her rapier (`DrawRapier`), and then use the rapier to `Stab` Rapier Roy.

With late commitment, we can abstain from thinking about whether Anne should have a rapier at the start of the story. Instead, the `CarryRapier` framing operator of Fig. 2 is introduced, transforming the fact that Anne is carrying a rapier into a mere possibility. As such, the operator can be used to satisfy the precondition of `DrawRapier` specifying that Anne must be wearing a rapier, in effect making the rapier necessary from a narrative point of view by the swordfight.

We can push late commitment further and place the requirement that Anne owns a rapier in a framing operator too (`OwnRapier`), with preconditions that the owning character is a pirate (pirates own swords), and there is a sword that is not owned by someone else (a sword cannot have two owners). The resulting plan can be seen in Fig. 3. Although the existence of rapiers is still introduced in the setting, the properties relevant for the outcome of the particular simulation (i.e., where they are, or who owns them) can be left unspecified at author-time, opening up a space of potential stories in which characters might have to fetch or even steal swords from each other to achieve their plans. For goal adoption, the question who hates who can be left to the simulation using the aforementioned `HateYou` framing operator to establish it.

4.2 Impro-POP: planning with IC and OOC operators

We introduce Impro-POP: an adaptation of partial order planning (POP) for interweaving IC and OOC operators. We have adapted the algorithm described in [12, ch. 11]; see Algorithm 1. We extended the `Choose-Operator` procedure of the planning algorithm by making a semantic distinction between operators whose selection is motivated IC (e.g., normal character actions) and OOC (e.g., framing operators or unintentional events). We add the option of choosing OOC operators in a failure-driven fashion; OOC operators are considered only after trying to reuse plan steps already in the (partial) plan or new (IC) actions. The algorithm is executed in an iterative deepening manner: it explores the full plan space for a given maximum number of plan steps (starting at 1), incrementing this number if it cannot find a plan at this depth.

A problem with combining IC and OOC operators in the planning process is that the planner might select IC operators in order to fulfil preconditions of OOC operators. This can lead to strange situations in terms of observed character motivation, such as a sailor hating his wife and planning to become a pirate, just so he can happen to be carrying a sword to stab her with. The action to become a pirate has no apparent IC motivation, and should only be allowed if the action can be justified for other reasons (e.g., the sailor seeks adventure). In terms of a partial order plan, we only allow IC operators to fulfil preconditions of OOC operators, if these actions also fulfil preconditions of other IC operators (establishing an alternative, coincidental IC reason for the action to exist in the plan). A practical implementation of this constraint is to constrain `Choose-Operator` so that an open precondition of an OOC operator is only fulfilled using new OOC operators, or any step that was already in the plan.

When allowing the planner to use framing operators, a second constraint is needed in the planner to make sure that the effects of a framing operator never contradict preconditions of action steps that might be executed earlier than the framing operator, since it must seem as if these effects have been the case from the beginning of the story. The framing operator should always be executed *before* the action whose preconditions it contradicts. For instance, if a `Parley` action (negotiate with the enemy) has a precondition that specifies that the pirate is not carrying a rapier (for safe negotiation), there is a contradiction with the effect of `CarryRapier` specifying that the character *does* carry a rapier. The agent may use `CarryRapier` in the same plan if it also includes an action in which the character puts down the weapon, but `CarryRapier` may not be ordered after `Parley` (establishing the absence of a rapier). To implement this constraint, we force the planner to pretend that it needs the start state to establish the effects of the framing operator. This is done by adding a causal link from the start step to the framing operator for each of these effects, so that the standard POP threat resolution will take care of the correct ordering.

With these constraints, the planner can believably include framing operators whenever they are useful for creating a plan. To execute the example plan of Fig. 3, the effects of the framing operators are translated into setting information that all the character agents receive and put in their knowledge base.

The planner is also used for the justification of a character goal, in order to find a way to establish the preconditions of the goal. The planning process is similar to that for action selection. The only difference is that a plan to justify a goal may not contain any IC operators, because the intention to establish the goal's preconditions is OOC. Therefore, the planner will only use OOC operators. After executing the resulting plan, the preconditions of the goal are met and the character can adopt the goal.

5 Discussion

We are currently successfully using late commitment in several small story domains, by replacing static parts of the setting by framing operators. Early experiments look promising; the framing operators are being used at sensible places in the story generation process. The challenge is to deal with the consistency issues that the use of late commitment raises. The preconditions of a framing operator help to prevent many inconsistencies, but private consistency checking by one agent is ultimately not enough. We discuss here three consistency issues that must be considered when using late commitment.

The first issue involves the impact that the execution of a framing operator has on characters other than the one that executes it. Since framing operators build on a shared reality, all characters must be able to unconditionally accept all effects of the framing operator as setting information, otherwise the operator should be aborted. The character agents have a mechanism to get approval from all the other characters before executing a framing operator. Currently, the other characters always accept; the decision whether to accept or reject a framing operator needs further exploration. The other characters have to check if a proposed framing operator is consistent with their personal world view; if it is not, they should refuse it unless they are able to justify the inconsistencies (e.g., "I only *thought* that there was no more rum, but forgot about that one bottle."). Since the execution of a framing operator leads to setting changes that are 'new' only from an OOC perspective, there are implications for the agents' design of the IC cognitive processes. The agents must be able to believably bypass these processes and adopt the new setting information directly as 'old' knowledge, or maybe believably accommodate the new information.

A second issue involves the history of the event sequence so far. IC operators executed in the past might have caused commitments to the setting, if their preconditions were fulfilled using setting information. For instance, the `Parley` example action of Sect. 4.2 commits to the fact that the agent carries no rapier, and it should disallow the character to frame a rapier to fight its enemy with when the negotiation goes sour later in the simulation. Impro-POP disallows plans in which an effect of a framing operator contradicts preconditions of actions earlier in the plan, but this does not capture the constraint that framing operator effects should also not contradict preconditions of actions that were already executed, if these preconditions were fulfilled by the setting at the time of execution.

A final issue concerns the presentation of the story (e.g., in natural language, or graphics). An assumption underlying late commitment is that the setting knowledge that is not yet defined in the system (and thus can be framed), is also not communicated to the reader or interactor of the story. While this assumption

Algorithm 1 The Impro-POP algorithm

function IMPRO-POP(*goal, operators*) **returns** *plan*
  plan ← MAKE-MINIMAL-PLAN(*goal*)
  loop do
    if SOLUTION(*plan*) **then return** *plan*
    S_{need}, c ← SELECT-SUBGOAL(*plan*)
    CHOOSE-OPERATOR(*plan*, *operators*, S_{need}, c)
    RESOLVE-THREATS(*plan*)
  end

function SELECT-SUBGOAL(*plan*) **returns** S_{need}, c
  pick a plan step S_{need} from STEPS(*plan*)
    with a precondition c that has not been achieved
  return S_{need}, c

procedure CHOOSE-OPERATOR(*plan*, *operators*, S_{need}, c)
  choose a step S_{add} that is either
    the start step S_{start} (**if** QUERY(c))
    a new step with c as effect for which holds:
      $S_{add} \in$ STEPS(*plan*)
      $S_{add} \in$ *operators* $\wedge$ IN-CHARACTER(S_{new}) $\wedge$ IN-CHARACTER(S_{need})
      $S_{add} \in$ *operators* $\wedge$ OUT-OF-CHARACTER(S_{new})
  if there is no such step **then fail**
  add the causal link $S_{add} \xrightarrow{c} S_{need}$ to LINKS(*plan*)
  add the ordering constraint $S_{add} \prec S_{need}$ to ORDERINGS(*plan*)
  if S_{add} is a newly added step from *operators* **then**
    add S_{add} to STEPS(*plan*)
    add $S_{start} \prec S_{add} \prec S_{finish}$ to ORDERINGS(*plan*)
  if S_{add} is a framing operator **then**
    for each $e \in$ EFFECTS(S_{add}) **do**
      add $S_{start} \xrightarrow{e} S_{add}$ to LINKS(*plan*)

procedure RESOLVE-THREATS(*plan*)
  for each S_{threat} that threatens a link $S_i \xrightarrow{c} S_j$ in LINKS(*plan*) **do**
    choose either
      Promotion: Add $S_{threat} \prec S_i$ to ORDERINGS(*plan*)
      Demotion: **if** S_{threat} is not a framing operator **then**
        add $S_j \prec S_{threat}$ to ORDERINGS(*plan*)
    if not CONSISTENT(*plan*) **then fail**
  end

function OUT-OF-CHARACTER(S)
  if S is an event schema or a framing operator **then return true**
  else return false

function IN-CHARACTER(S)
  return $\neg$ OUT-OF-CHARACTER(S)

applies to a large extent to textual representations, it may apply much less to graphical representations (in which showing a pirate also shows whether he is carrying a rapier or not), needing explicit consideration for what is exactly being communicated. Framing operators that deal with the domain of interpersonal relationships ("Let's pretend I was your father") or backstories ("Let's pretend you have hurt me in the past") may be less problematic, since they are often conveyed through dialogue.

6 Conclusions and Future Work

The technique of late commitment described in this paper allows virtual agents to make run-time commitments to previously undefined aspects of a virtual story world using operators that we call *framing operators*. We have introduced Impro-POP: an adaptation of a standard partial order planning algorithm that also considers framing operators (and other OOC operators such as events) when resolving open precondition flaws, whilst ensuring that an agent never selects actions IC with the OOC intention of making framing operators possible. Impro-POP facilitates plan construction for action selection and enables the agent to justify the adoption of new goals. The benefit of late commitment is that it takes away the responsibility of the author to predict which exact properties the story world needs to have for a particular course of the simulation. Furthermore, it offers the characters more flexibility in their reasoning processes; they can to a certain extent *choose* to adopt goals and enable actions, by filling in the world around them.

We have described some of the consistency issues that were raised by our first experiments with the use of framing operators in the Virtual Storyteller. Further experimentation will have to point out if there are more, how fundamental they are, and how they can be solved.

The use of framing operators is certainly not limited to action selection and goal justification. We aim to investigate what other processes can benefit from the use of late commitment. For instance, framing operators might enable perceptions (e.g., discovering a hidden door) or facilitate emotional reactions. These investigations are in service of the longer-term aim to develop criteria for story progression that allow the agents to take a more proactive stance towards story development (e.g., try to have goals that 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). Thanks to Edze Kruizinga for implementing the planner.

References

1. Mateas, M., Stern, A.: Towards integrating plot and character for interactive drama. Working notes of the Social Intelligent Agents: The Human in the Loop Symposium (2000)
2. Riedl, M.O.: Narrative Generation: Balancing Plot and Character. PhD thesis, North Carolina State University (2004)
3. Aylett, R., Louchart, S., Dias, J., Paiva, A., Vala, M., Woods, S., Hall, L.: Unscripted narrative for affectively driven characters. IEEE Computer Graphics and Applications **26**(4) (2006) 42–52
4. Swartjes, I., Vromen, J.: 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)
5. Louchart, S., Kriegel, M., Figueiredo, R., Paiva, A.: Authoring emergent narrative-based games. Journal of Game Development **3**(1) (2008) 19–38
6. Louchart, S.: Emergent Narrative - towards a narrative theory of Virtual Reality. PhD thesis, University of Salford, Salford, UK (2007)
7. Swartjes, I., Theune, M.: The virtual storyteller: Story generation by simulation. In: Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC). (2008)
8. Swartjes, I., Kruizinga, E., Theune, M.: Let's pretend I had a sword: Late commitment in emergent narrative. In: Proceedings of the First Joint International Conference on Interactive Digital Storytelling (ICIDS '08). (2008)
9. Swartjes, I., Kruizinga, E., Theune, M., Heylen, D.: Emergent narrative and late commitment. In: Proceedings Intelligent Virtual Agents. (2008)
10. Garrett, Y.: Offer, accept, block, yield: The poetics of open scene additive improvisation. Master's thesis, University of Sydney (2006)
11. Riedl, M., Young, M.: Story planning as exploratory creativity: Techniques for expanding the narrative search space. In: Proceedings of the 2005 IJCAI Workshop on Computational Creativity. (2005)
12. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (1995)