Character-Focused Narrative Generation for Execution in Virtual Worlds

Mark O. Riedl and R. Michael Young

Liquid Narrative Group, Department of Computer Science, North Carolina State University, Raleigh, NC 27615, USA.
`moriedl@ncsu.edu, young@csc.ncsu.edu`
`http://liquidnarrative.csc.ncsu.edu/`

Abstract. Because narrative plays such a key role in the understanding of events in our daily lives, the ability to generate narrative can be of great use in virtual reality systems whose purpose is to entertain, train, or educate their users. Narrative generation, however, is complicated by the conflicting goals of *plot coherence* – the appearance that events in the narrative lead towards the narrative's outcome – and *character believability* – the appearance that events in the narrative are driven by the traits of the story world characters. Many systems are capable of achieving either one or the other; we present a new approach to narrative generation in the Actor Conference system, which is capable of generating narratives with both plot coherence and character believability. These narratives are declarative in nature, readily lending themselves to execution by embodied agents in virtual reality environments.

1. Introduction

Narrative as entertainment, in the form of oral, written or visual stories, plays a central role in our social and leisure lives. There is also evidence that we build cognitive structures of the real events in our lives represented as narrative to better understand what is happening around us [4]. This "narrative intelligence" is central in the cognitive processes that we employ across a range of experiences, from entertainment contexts to active learning. Interaction within a virtual world, especially one created by an interactive 3D virtual reality system, provides an engaging environment in which a system's user can readily view unfolding action as narrative. In narrative-oriented virtual reality systems, the effectiveness of interaction is enhanced when the actions of the system-controlled characters – intelligent computer agents embodied as graphical avatars – are controlled and coordinated with one another to provide a coherent storyline.

The ability to structure system characters' action sequences so that they can be understood as elements of a story's narrative is of importance to systems that wish to effectively use narrative for entertainment, training, or education. Most existing narrative-oriented virtual worlds are built using pre-scripted action sequences; characters play out the same elements of the same story each time the system is run. In contrast, a system that generates a novel narrative structure for each user session can tailor its narratives to the individual preferences or needs of the user instead of relying on scripted sequence prepared in advance. Automatic narrative generation presents many technical challenges, however, one of which is the ability to balance the trade-offs between *plot coherence* and *character believability*. A plot (or action sequence) is coherent when a user can understand the way in which the events of a narrative have meaning and relevance to the outcome of the story. A character is believable [3] when the actions taken by the character can be seen to come from the character's internal traits. While narrative coherence is essential for an audience to make sense of what they are seeing, character believability is important in a virtual reality medium where characters are expected to be expressive and entertaining.

The research presented here considers the importance of the trade-off between plot coherence and character believability. In general, narrative generation systems that generate highly coherent narrative structures often neglect issues of character and believability. Likewise, systems that capitalize on the use of highly believable characters tend to promote poor narrative structure. In this paper, we present the narrative generation system, the Actor Conference (ACONF), which attempts to address the weaknesses and capitalize on the strengths of the various existing approaches to automated narrative generation.

2. Related Work

In order to comprehend the relevance of related work to our own research, we present a framework for categorizing narrative generation systems. The classification framework ranks narrative generation systems along two continuous dimensions: plot coherence and character believability. The ideal situation is to be able to generate narratives that are both high in plot coherence and character believability.

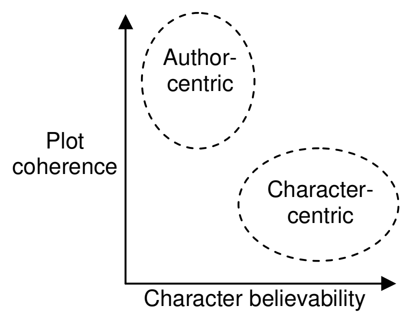

Fig. 1. Classification framework for narrative generation systems.

Story-generation systems can also be categorized as *author-centric*, *story-centric*, or *character-centric* [8] (adapted from [2]). Author-centric systems model the thought processes of an author. Story-centric systems model linguistic and grammatical properties of story texts, such as in [12] and [2]. Character-centric systems model the goals, beliefs, and plans of characters in the story-world with the intention that story emerge as characters pursue autonomous goals and interact with each other. The two taxonomies are tightly coupled; we believe that character-centric systems tend to result in stories with strong character-believability but weak plot coherence, while author-centric systems result in stories with strong plot coherence but weak character believability. We do not consider story-centric systems further because they often do not utilize strong notions of plot or character, focusing instead on discourse structure of storytelling. The classification framework is shown in Figure 1.

Character-centric systems rely on the concept of emergent narrative [1], which postulates that narrative emerges from unstructured interaction of autonomous agents. Narrative arises from the interaction between agents, similar to the way story can emerge through free improvisation or through structured activities such as game playing. Because emergent narrative relies on interactions, these systems can capitalize on the use of animated agents that contain a rich repository of behaviors and emotions. One of the risks of emergent narrative, however, is that narrative may not emerge [1]. This fragility is weighed against believability of the experience; when narrative emerges, the user will be engaged with a rewarding experience. Tale-spin [9] explicitly represents characters as collections of beliefs, needs, and interpersonal relationships. An inference engine determines how the characters could behave in order to achieve their needs and narrative emerged from the interactions chosen by the inference engine. The Oz project [7;3] situates a user in a virtual environment populated by autonomous, animated agents. Each animated agent has a set of goals and beliefs and autonomously works towards achieving its personal goals. In order to ensure an interesting experience for the user, an external module – a drama manager – attempts to discretely manipulate the autonomous agents' desires in order to force narrative to emerge. Interactive narrative – a focus of the Oz project – is a special sub-problem in narrative generation, but [14] demonstrates a similar approach to narrative generation that does not involve an interactive user.

In contrast to character-centric systems, author-centric systems involve computational theories for generating narratives. These systems algorithmically piece together a narrative as a sequence of events that are related through some form of logical structure. Since author-centric systems generate narrative through a structured, rational methodology they are not plagued by failure in the same way that character-centric systems are. However, by focusing on the logical structure of a narrative, character actions making up the events in a narrative will be chosen to fit the narrative's structure and not necessarily chosen because that is the natural course of action for a believable character to take. The Universe system [5;6] uses a planner to select a sequence of actions for the characters in the story world to perform. The planner in Universe only incorporates actions into the narrative sequence that contribute to the system goals although system goals may be described at a high level of abstraction, such as "keep lovers apart" (Universe operates in the domain of soap-operas). Defacto [13] uses a rule-based approach to narrative generation. A knowledge base is populated with rules about character relationships, goals, social norms, as well as rules about intention and the attempt to perform actions. The rules are encoded in a format which enables the system to reason logically about character intentions and actions. The result of narrative generation is a list of temporally ordered attempted actions that are assigned success or failure in order to achieve an outcome that is satisfying and suspenseful.

3. The Actor Conference System

The Actor Conference (ACONF) system is explicitly designed to take advantage of the strengths of both the character-centric and author-centric techniques and thus achieve both strong plot coherence and strong character believability. ACONF is itself an author-centric system and, like the Universe system [5;6], uses a decompositional, partial-order planner to assemble a sequence of actions, comprising the narrative. The actions in the plan represent the behaviors that the characters are going to perform as part of the narrative. Using a planner for narrative generation is advantageous for two reasons. First, planners operate by identifying causal relationships between actions which naturally map to the domain of narrative [16]. Secondly, the output of a planner is a temporally ordered sequence of discrete operations. These operations can be directly executed by agents in the virtual world [17;11].

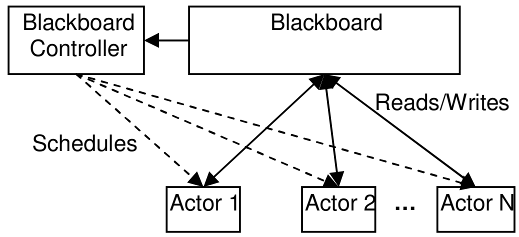

Fig. 2. The Actor Conference architecture.

Partial-order planners, however, are not alone adequate for the task of narrative generation. Consider the fact that a partial-order planner chooses only those actions that are necessary to achieve a certain world state. Thus, in a narrative, characters whose actions are planned by a partial-order planner will perform behaviors that will bring about that state but not necessarily perform behaviors that are consistent with the audience's expectations. Believable characters have idiosyncrasies and are expected to perform behaviors that are motivated by individualistic beliefs and desires instead of narrative structure, which they, arguably, would not be aware of.

In order to capture personalized character behaviors during conventional partial-order planning, we introduce expert systems for each of the characters that exist in the story world. Each expert system – referred to as an "Actor" – is instantiated with a full understanding of a single story world character, the actions it can perform, and the ways it will react to situations and interact with other characters. The expert system is not the character itself, but understands its assigned character as if it were an actor in a play who is tasked with giving life to its character. The responsibility of constructing a coherent narrative is thus distributed among the experts. Using a blackboard architecture, a single planning problem is broken up into smaller planning problems and handed off to each Actor so that it can plan a sequence of behaviors that achieve a particular goal in a believable fashion. The smaller planning problems are reassembled onto the blackboard and the process iterates. Actor expert systems are limited in their planning processes to only considering actions for the character that it represents. This limitation prevents one Actor from making decisions about other characters' actions and also provides convenient boundaries with which we can split up and distribute the narrative plan structure. A diagram of the ACONF architecture is shown in Figure 2.

3.1. Narrative Planning in Actor Conference

Since ACONF uses decompositional planning to model the narrative authoring process, narrative is represented as a partial-order, causal link plan where each step in the plan is an abstract or primitive action that a character in the story world will perform. The blackboard contains structures called hypotheses – guesses about the solution – that contain (possibly) incomplete narrative plans and sets of associated annotations. The blackboard provides an architecture for control and coordination, but narrative generation is in the hands of the experts. The experts – Actors – are autonomous agents that represent individual characters in the story world and encapsulate the ability to plan actions that their characters perform in the story world. At the core of each Actor is the Longbow decompositional, partial-order planner [15].

Actor Conference builds a narrative as a single plan, consisting of actions performed by all characters in the story world. A single plan that will control the performance of every character is useful for generating narratives with strong plot coherence because the plan actions will be guaranteed to be causally relevant to the outcome of the narrative [11]. The two issues that our research addresses are: (a) how a single narrative plan can be distributed among many agents and then reassembled, and (b) how multiple Actor agents can utilize the well-established paradigm of partial-order planning for highly characterized action planning.

3.1.1. Cast Calls

Deferring for now the issue of how Actor agents can plan character-specific actions, we address how the single narrative plan can be broken down and distributed to Actor agents and reassembled. An Actor receives a hypothesis from the blackboard, containing a narrative plan. The narrative plan is incomplete when it contains one or more flaws, such as an action with unsatisfied preconditions or an abstract action that has not been expanded into less abstract actions. Plan flaws are resolved through iterations of the planning algorithm [10;15]. For now, let us assume that an Actor, *A*, representing story world character, Kate, has been scheduled to refine a hypothesis containing an incomplete plan. Actor *A* is tasked with resolving flaws in the narrative plan by placing actions into the plan structure that best illustrate Kate's character.

Unless ACONF is generating a one-man play, an Actor is invariably going to have to incorporate the actions of other characters into its plan. To handle the situation of character interaction, we employ modifications to the standard planning process. First we encourage the use of highly hierarchical plan structures. This gives us two advantages. The first advantage is that hierarchical plans can be constructed at different levels of abstraction that help define the structure of narrative and can guide the Actor as it refines the plan. The second advantage is that, at a sufficiently high level of abstraction, characters do not exert idiosyncratic behavior. Suppose that Actor *A* inserts the action, `talk-about(Joe, Kate, sports-cars)` – in which the character, Joe, will speak to Kate about sports cars – into the narrative plan. `Talk-about`, as an abstract description of a communicative act, captures the essence of the interaction between Kate and Joe without concern that Joe might have a tendency to be long-winded when speaking on the subject of sports cars.

The planning algorithm used by the Actor agents, however, has been modified in the following way: it is prohibited from decomposing abstract actions that are to be performed by other characters. That is, when Actor *A* comes across a flaw that requires it to expand an abstract action that is not performed by Kate, Actor *A* is forced to leave the flaw unresolved, leaving a gap in the completeness of the plan. To continue the previous example, Actor *A* is able to insert the action, `talk-`

`about(Joe, Kate, sports-cars)`, into the narrative plan (assume that `talk-about` establishes some condition later in the plan that Kate understands sports-cars), but, because it describes an action to be performed by Joe, Actor *A* leaves the abstract action unexpanded. We refer to these gaps as *cast calls* because they are points in the narrative plan where other Actors can script themselves into the story. When an Actor posts a hypothesis to the blackboard containing a further refined, yet still incomplete plan, it is analyzed by the blackboard controller for cast calls. One or more Actors are scheduled to respond to the cast call, retrieve the hypothesis from the blackboard, and begin refining the plan contained within. Presumably any Actor that responds to the cast call identifies with the character described in the cast call and can therefore expand the abstract action left by the previous Actor.

The question remains of who gets to respond to the new hypothesis once it is posted to the blackboard. Certainly the Actor representing Joe is a candidate to refine the hypothesis. However, as far as the creator of the hypothesis, Actor *A*, is concerned, the flaw need only be resolved by a character that fills the same role that Joe plays, e.g. some character knowledgeable about sports cars. Therefore, when the cast call is created, the plan is annotated with a description of a character role, and not a specific character name. The role is determined by analyzing the preconditions and constraints of the unexpanded action.

3.1.2. Actor Planning

Narrative planning occurs in manageable chunks inside the Actor expert systems. Each actor is an expert on a single character in the story world and is motivated to choose the behaviors for that character that best illustrate the character's traits within the constraints of the unfolding plot. Since an Actor performs planning as a response to a cast call, all actions that are planned are either to be performed by the character represented by the Actor or are high-level descriptions of interaction between characters. Character expertise in the Actor agent is captured in two different ways. First, each Actor agent has its own action library that defines the actions and decomposition rules that a single character can perform. Second, each Actor captures character through a customized plan search heuristic function.

The Actor's private action library can be thought of as a knowledge-base describing how an individual story world character behaves and interacts with the world. The action library contains a complete set of actions that the character can perform. Each action has a specification of the conditions that need to be true for the character to perform the action and the way in which the action affects the state of the story world once it is performed. Most Actors will share some similarity in the actions in their action libraries; however characterization relies on the ability of actions to differ in their preconditions and effects. For example, violent actions in a moral character's action library may require that the character believe the victim of the violence to be deserving of the outcome.

Furthermore, some actions are designated as primitive and others are designated as abstract. The planner attempts to expand abstract steps, under the restrictions described in the previous section, by applying decomposition rules to instantiated abstract actions. One can think of these decomposition rules as schemata for how the character will behave. Decomposition rules also allow for idiosyncrasies to be expressed because decomposition rules can describe subsequences that contain actions that are not rationally motivated. It is possible, and even desirable, for there to be more than one decomposition rule for every abstract action the character can perform. However, there does not have to be abstract actions for every possible circumstance that a character might find herself in. When an Actor lacks an abstract action to capture a circumstance, it can rely on the basic planning algorithm to insert plausible sequences of customized, primitive actions into the plan.

Besides customized action libraries, Actors capture character expertise through the use of customized plan-space search heuristics. The Longbow planner uses search heuristics to utilize domain knowledge to perform a best-first search through the space of possible plans [15;10]. In ACONF, the planning process is distributed among the Actor agents and for each Actor the planning domain is the story world character itself. The heuristic function for each Actor captures its character's preferences about the types of actions it likes to perform and the elaborateness of the sequence.

3.2. From Plan Space to Hypothesis Space

Actors search for plans within the space of all possible, sound plans [10;15]. The ACONF system, as a collection of collaborating agents, searches for a complete hypothesis in the space of all possible hypotheses. As an Actor searches for an incomplete but sound plan, it necessarily leaves regions of the plan space unexplored; an Actor cannot explore the entire plan space due to complexity trade-offs and due to prohibitions from considering actions for characters other than the one it represents. However, there may be many possible candidate plans that the Actor can find. This is especially true if the Actor is expanding an abstract action and has more than one applicable decomposition rule in its action library. If the Actor commits to a plan, it is committing to a particular structure for the narrative and this commitment will guide how the other Actors in the system refine and construct their own hypotheses. This raises the issue of plan space backtracking. Each Actor is only solving a localized portion of the overall problem; what may seem valid in the local scope may have severe repercussions to the system as a whole as other Actors could be left unable to refine the solution. Since each Actor searches the plan space independent of the others, one can think of each hypothesis as having its own plan space. There is no way for one Actor to backtrack to a part of the overall plan space that another Actor chose not to explore. This separation of plan spaces is shown in Figure 3.

In Figure 3, there are three hypotheses in the hypothesis space on the blackboard: *X*, *Y*, and *Z*. Actor *A* posts hypothesis *X* to the blackboard and, during the process of creating hypothesis *X*, explores a portion of the plan space. The plan space is shown as the tree structure inside the hypothesis. Each smaller circle is a plan node in the plan space. Circles that are crossed out represent inconsistent plans. The dashed triangles represent branches of the plan space that have not been explored. The double-lined circle represents the plan that Actor *A* commits to (a plan is sound and complete except for decomposition of abstract steps to be performed by other characters). Actors *B* and *C* both attempt to refine hypothesis *X* but cannot, for whatever reasons, find plans that resolve the flaws that Actor *A* left behind. If there are no alternatives to hypothesis *X*, narrative generation will fail! It is possible that

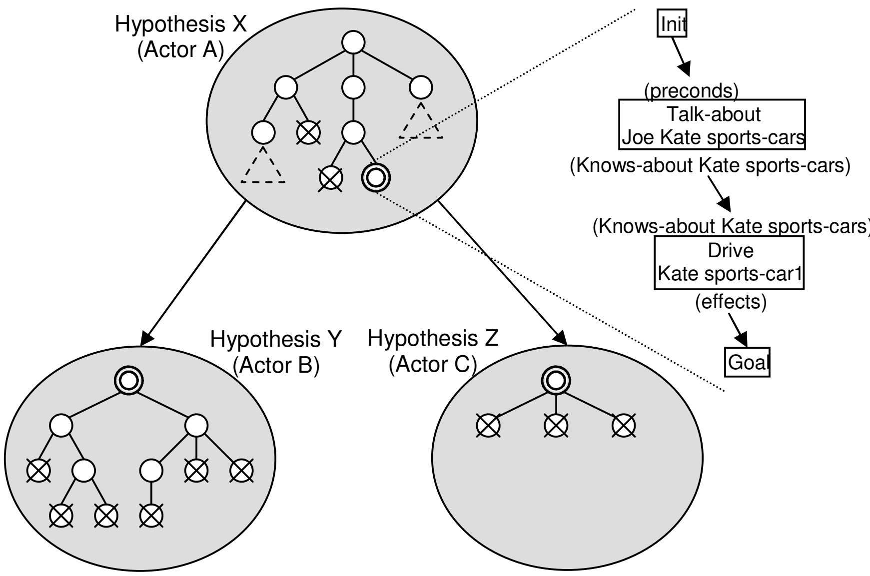

Fig. 3. Plan spaces within the hypothesis space. The gray circles are hypotheses. The smaller, embedded, circles are nodes in the narrative plan search space. Each plan space node contains a (possibly flawed) plan, one of which is shown expanded to the right.

another plan exists in the unexplored regions of hypothesis *X*'s plan space, but Actors *B* and *C* are helpless to explore these regions because it is part of a different Actor's decision-making process. Because the hypothesis space is unrelated to plan space, we are threatened by the possibility that narrative generation in ACONF is incomplete.

Incompleteness for this reason is unacceptable. Therefore, we have modified the blackboard controller to allow hypotheses to be *revisited*, that is, for an Actor to discard the plan it previously committed to and search for a new partial solution in a previously unexplored region of the plan search space. Revisitation should not be confused with backtracking in the hypothesis space. Backtracking in hypothesis space means to choose an alternative hypothesis to expand. For example, when hypothesis *Y* is found to be inconsistent in Figure 3, ACONF backtracks to try an alternative: hypothesis *Z* (which is also found to be inconsistent). However, it is clear that the only hope of finding a complete narrative in the example in Figure 3 is to revisit hypothesis *X* and to expand regions of the plan space contained within. Just as a partial-order planner maintains a queue of unexplored plans in plan space [10], the blackboard controller maintains a queue of unexplored hypotheses in hypothesis space. In order for hypothesis revisitation to work, when a hypothesis is explored, it is not removed from the queue. Instead it is re-ranked (we assume the blackboard controller is searching the hypothesis space in a best-first manner) and reinserted into the queue. With revisitation, the hypothesis search problem in ACONF is as complete as the partial-order planner used by the Actors in the system.

4. Narrative Plan Execution and Interactivity

ACONF is a purely generational system and does not handle the execution of its narrative plans. Instead, ACONF can be coupled with an execution engine, such as Mimesis [17;11], that is specifically designed for execution of partial-order plans in a 3D graphical virtual world. The Mimesis architecture consists of a narrative generation component, responsible for dynamic generation and repair of narrative plans, and an execution substrate, responsible for performing the narrative plan through systematic control of animated avatars in a 3D virtual world. In this case, the narrative generation component encapsulates the ACONF system.

Complications to the plan execution process arise in two ways. The first is due to a mismatch in the characterization of actions used by a partial-order planner, such as ACONF, to produce plans and the code used by the virtual world to implement them. The story plan is a sequence of discrete actions whose effects happen instantaneously. In contrast, most of the corresponding character behaviors in the virtual world require a relatively long period of time to complete. This is especially true when actions must be coordinated with slower graphical character animations. To preserve the temporal semantics of the plan's structure, the Mimesis execution substrate launches sentinel threads for each action to be performed [17]. These sentinel threads monitor the environment until the effects of the action have been achieved or the action has failed.

The second complication dealt with during plan execution arises when a user assumes the control and identity of a story-world character through its avatar. Given that the user is not constrained to follow the narrative plan and may be only partially aware of the narrative plan, that character is now capable of performing actions within the story world that interfere with the structure of the story plan [11]. Because each action that the user performs might change the world state in a manner that would invalidate some unexecuted portion of the narrative plan, the Mimesis execution substrate monitors the user's action commands and signals an *exception* whenever the user attempts to perform an action that would change the world in a way that conflicts with the causal constraints of the story plan. Exceptions are handled by replacing the current narrative plan with a contingency plan. Contingency plans are pre-computed by Mimesis through repeated calls to ACONF with varied initial conditions. The way in which exceptions are handled within the Mimesis architecture is described further in [17] and [11].

5. Conclusions

Actor Conference is a narrative generation system that utilizes techniques from author-centric and character-centric narrative generation systems in order to balance the conflicting concepts of plot coherence and character believability and generate narratives that are both apparently understandable and character-driven. The system generates narrative plans – partially-ordered sequences of story world character actions – with rich temporal and causal structure. The causal nature of the narrative plans ensures plot coherence because character actions establish the conditions necessary for the narrative goals of the story. Character believability is achieved by distributing the partially built narrative plan structures to expert agents that represent characters in the story world. With a slightly modified planning process, highly customizable action libraries, and heuristic functions than rank the believability of a sequence of actions, the expert "Actor" agents are able to illustrate the traits of the characters they represent, despite the rational nature of planning.

ACONF uses blackboard architecture coordinates the efforts of the numerous Actor agents, effectively making its narrative generation process a search through the space of hypotheses, or partial narratives. Each hypothesis in the search space contains a fragment of the overall narrative plan search space. Heuristics inform the process of search through the hypothesis search space, allowing for the possibility of revisiting previously explored hypotheses so that planning completeness is assured.

References

1. Aylett R. Narrative in virtual environments – towards emergent narrative. In: *AAAI Fall Symp. on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
2. Bailey P. Searching for storiness: story generation from a reader's perspective. In: *AAAI Fall Symp. on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
3. Bates J. The role of emotion in believable agents. In: *CACM* 1994; 37; 122-125.
4. Bruner J. *Acts of Meaning*. Harvard University Press: Cambridge, MA 1990.
5. Lebowitz M. Creating characters in a story-telling universe. *Poetics* 1984; 13; 171-194.
6. Lebowitz M. Story-telling as planning and learning. *Poetics* 1985; 14; 483-502.
7. Mateas M. (1997). An Oz-centric review of interactive drama and believable agents. *Technical Report CMU-CS-97-156*. School of Computer Science, CMU, Pittsburgh 1997.
8. Mateas M, Sengers P. (1999). Narrative intelligence. In: *AAAI Fall Symp. on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
9. Meehan J. Tale-spin, an interactive program that writes stories. In: *Proceedings of the 5th International Joint Conferences on Artificial Intelligence*. 1977.
10. Penberthy J, Weld D. UCPOP: A sound, complete, partial order planner for ADL. In: *Proceedings of the 3rd Int. Conference on Knowledge Representation & Reasoning*. 1992.
11. Riedl M, Saretto C, Young, R. Managing interaction between users and agents in a multi-agent storytelling environment. In: *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*. 2003.
12. Rumelhart D. Notes on schema for stories. In: *Representation and Understanding: Studies in Cognitive Science*. Bobrow D, Collins A eds. 1975.
13. Sgouros N. Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence* 1999; 107; 29-62.
14. Theune M, Faas S, Nijholt A, Heylen D. The virtual storyteller: Story creation by intelligent agent. In: *Proceedings of the First International Conference for Interactive Digital Storytelling and Entertainment*. 2003.
15. Young R, Pollack M, Moore J. Decomposition and causality in partial-order planning. In: *Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems*. 1994.
16. Young R. Notes on the Use of Plan Structures in the Creation of Interactive Plot. In: *AAAI Fall Symp. on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
17. Young R, Riedl M. Towards an architecture for intelligent control of narrative in interactive virtual worlds. In: *Proceedings of the Int. Conference on Intelligent User Interfaces*. 2003.