Character-focused Narrative Planning

MARK O. RIEDL, R. MICHAEL YOUNG

Liquid Narrative Group

Department of Computer Science

NC State University

Raleigh, NC 27695

+1.919.513.3038

moriedl@ncsu.edu, young@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.

Keywords: *Narrative generation, Believable agents, Plot coherence, Planning, Blackboard architecture*

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 [6]. 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 seqeunces 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 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 especially important in a virtual reality medium where characters are expected to be highly expressive and entertaining.

Narrative theorists typically structure narrative into three layers: the fabula, the story, and the text [4]. The *fabula* refers to a complete characterization of all the events occurring in the story world, along with the specification of the chronological order in which they occur. The *story* is the subset of the fabular elements selected by the author to be included in the narrative. The *text* is the result of rendering the story into a medium of communication (e.g., text, film, 3D graphics). For the purposes of the research presented here, when we refer to narrative generation we specifically mean fabula generation. The fabula is the most basic aspect of narrative and its structure is centered around the concepts of action and causality – concepts which correspond to well understood models of computation [20]. For virtual reality systems, the fabula can be rendered directly into action specifications for execution by embodied agents. In this paper, we will assume some other process will be used to generate the story and text levels of the narrative from the fabula. Futher, we will focus only on the generation of narrative compose of the actions of system-controlled characters, ignoring the issue of user interaction within our narratives; see [14; 21] for detailed descriptions of techniques for managing user interaction with the narrative structures we describe here.

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, which attempts to address the limitations and capitalize on the strengths of the various existing approaches to automated narrative generation. Section 2 gives an overview of related work and a classification framework entailing the relative strengths and weaknesses of various approaches to the problem of narrative generation. The framework also provides motivation for the narrative generation techniques utilized in Actor Conference. Section 3 describes how Actor Conference generates character-focused narratives.

2. Related work

In order to comprehend the relevance of related work to our own techniques, 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. Narratives are said to be *coherent* when the audience perceives that events in the narrative lead toward an outcome. A coherent plot can be summarized in relatively few sentences [17], indicating that the narrative is structured around a few key concepts. This metric is one shared both by the Disney story team and by Alfred Hitchcock [17]. *Character believability* is the extent to which characters in the narrative exhibit rich personalities, emotion, social behavior, motivations, and personal goals [10]. Thus a narrative with believable characters is one in which the action appears to fall out of character interactions. The ideal situation is to be able to generate narratives that are both high in plot coherence and character believability.

Story-generation systems can also categorized as *author-centric*, *story-centric*, or *character-centric* [11] (adapted from [3]). Author-centric systems model the thought processes of an author. Story-centric systems model structural and grammatical properties of story texts. 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. This is also referred to as emergent narrative [2]. 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-

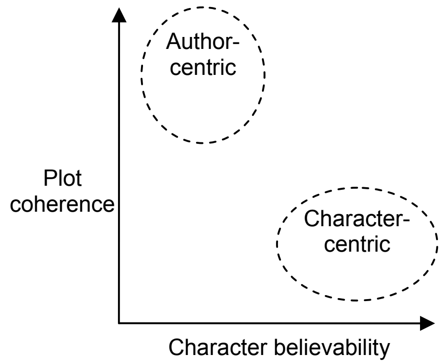

Figure 1. Classification framework for narrative generation systems.

centric systems further because they focus solely on grammatical and linguistic constraints of story. The classification framework is shown in Figure 1.

2.1. Character-centric systems

Character-centric systems rely on the concept of emergent narrative [2] which postulates that narrative can emerge from the 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 [2]. Because emergent narrative relies on interactions, these systems can capitalize on the use of animated agents that are very expressive and contain a rich repository of behaviors and emotions. The use of behavioral animated agents has the positive effect of making the interactions both believable and engaging [5]. One of the risks of emergent narrative, however, is that narrative may not emerge [2]. This fragility is weighed against believability of the experience; when narrative does emerge the user will be engaged with a rewarding experience.

Tale-spin [12] explicitly represented characters as collections of beliefs, needs, and interpersonal relationships. An inference engine determined how the characters could behave in order to achieve their needs and narrative emerged from the interactions chosen by the inference engine. Tale-spin was dependent on correct encodings of characters and their relationships; a knowledge base lacking the proper inference rules would result in the system's failure to generate a narrative that made sense [12].

The Oz project [10] 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, a drama manager attempts to identify situations, called plot points, which will lead to a narrative experience. The drama manager discretely manipulates the autonomous agents' beliefs and goals such that they will work towards fulfilling the narrative goals even though this causes some inconsistency [10; 5]. The OZ project is an interactive drama system, meaning a user enters the virtual world and interacts with the characters and affects the plot. We regard interactive drama systems as a specialization of narrative generation but only consider the narrative aspects of the system here. [16] demonstrates a system with a narrative generation component that operates similar to that in the Oz project, but without an interactive user.

2.2. Author-centric systems

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. Author-centric systems generate narrative through a structured, rational methodology and are, hence, not plagued by failure in the same way as character-centric systems. However, by focusing on the logical structure of a narrative, issues of character and believability are often glossed over. Character actions, which make up the events in a narrative, will be chosen to fulfill 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 [8; 9] uses a planner to select a sequence of actions for the characters in the story world to perform. The system was initialized with a set of high-level goals, such as "cause a husband character and a wife character to divorce" (the Universe system operates in the soap-opera domain). The sequence of actions chosen by the planner will bring about the specified, high-level goals and thus obtain a high degree of plot coherence. Unfortunately, the planner in Universe only incorporates actions into the narrative sequence that contribute to the system goals. For example, a protagonist and an antagonist, who might be expected by the audience to engage in combat if their paths crossed, would not fight unless the system goals indicated that one of them should get hurt, regardless of whether their paths did cross.

Defacto [15] uses a rule-based approach to narrative generation. A knowledge base is populated with rules about character relationships, character goals, social norms, as well as rules about intention and the attempt to perform actions. The rules are encoded in a format very similar to first-order predicate calculus which enables the system to reason about character intentions and actions in method very similar to theorem proving. The result of narrative generation is a list of temporally ordered attempted actions. A user and the system collaborate to assign success or failure conditions to each attempted action in a way that is satisfying and suspenseful.

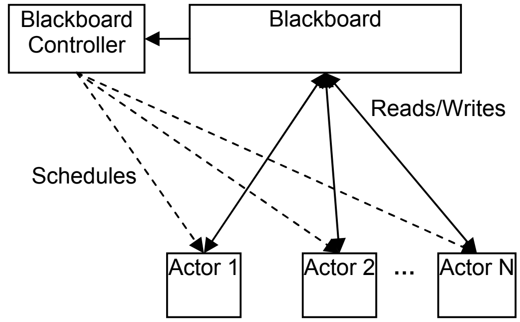

Figure 2. The Actor Conference architecture.

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 cohesion and strong character believability. ACONF is itself an author-centric system and, like the Universe system [8; 9], 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 [20]. 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 [21; 14].

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 goal. Thus, in a narrative, characters whose actions are planned by a partial-order planner will perform behaviors that will bring about a certain state of the world but not necessarily perform behaviors that are consistent with the audience's expectations. Believable characters have idiosyncrasies and are expected to perform non-task related behaviors because, arguably, they would not be aware of the narrative structure. For example, characters who are antagonists are expected by the audience to impede the protagonist's progress towards the goal of "happily ever after."

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. 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

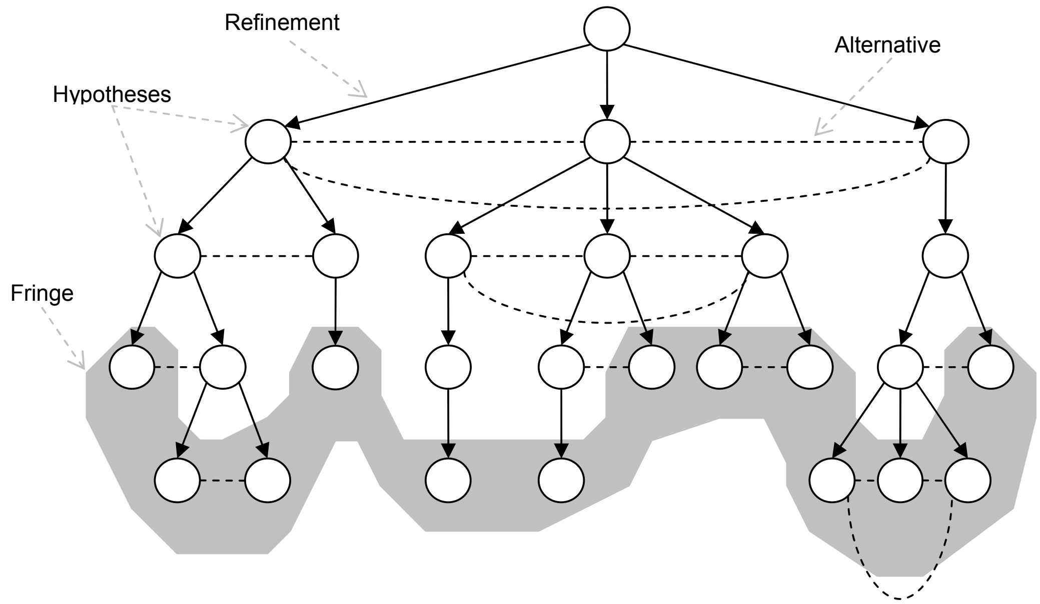

Figure 3. The hypothesis search space of the blackboard.

problems are reassembled onto the blackboard and the process iterates. The responsibility of constructing a coherent narrative is thus distributed among the experts. 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. Overview of blackboard architectures

The blackboard is a technique for managing collaboration among a large number of autonomous agents that are working together to solve a single problem [7]. Each agent is considered to be an expert at solving a small part of the over-all problem. The blackboard makes available a global representation of the problem in its current form and when an agent recognizes a part of the problem as one that it can solve, it modifies the representation. Together the autonomous agents incrementally work through the problem space towards a solution.

Blackboard architectures are typically broken up into a global problem space, a controller, and one or more expert autonomous agents [7]. The problem space contains many hypotheses – partial solutions – about the solution to the problem. The blackboard controller searches the problem space for places in which the registered expert agents can make contributions. Expert agents that are deemed eligible by the blackboard controller are scheduled for an opportunity to refine a single hypothesis revising the hypothesis and posting a new hypothesis representing an incrementally more complete solution. Each refinement will trigger other expert agents to make further refinements until a solution is found. The order in which hypothesis nodes are visited is determined by the blackboard controller. The controller observes the problem space as new hypotheses are posted. Newly posted hypotheses – those on the fringe of the hypothesis search space – are analyzed for patterns that could potentially interest any of the available knowledge-sources. Relevant expert agents are scheduled to make their refinements in a best-first fashion. One interesting feature of blackboard systems is that hypotheses can be visited in any order, so expert agents do not carry over information or experiences when transitioning from one hypothesis to another [7]. Thus hypotheses must be fully self-contained sub-problems and carry all the information a knowledge source needs to work on that particular sub-problem. An expert agent can, however, benefit from the experiences of other knowledge sources who have worked on the hypothesis previously through *annotations* attached to the hypothesis. For example, one knowledge source may provide direction for future refinements by leaving notes to be ignored opportunistically followed up on. Figure 3 shows an example of a hypothesis search space.

3.2. Narrative planning in ACONF

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. On the blackboard, each hypothesis is, therefore, an incomplete narrative as represented by a flawed plan. The blackboard provides 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 will perform in the story world. At the core of each Actor is the Longbow decompositional, partial-order planner [19].

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. 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.

3.2.1. 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 then reassembled. Consistent with our description of blackboard architectures, the hypothesis structure contains a (possibly) incomplete narrative plan and a set of annotations. The narrative plan is incomplete because 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 [19]. 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 the character of Kate.

Unless ACONF is generating a one-person 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 Joe and Kate without concern that Joe might have a tendency to be long-winded when speaking on the subject of sport-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 does not belong to 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 knows about sports-cars), but, because it describes an action to be performed by the character, 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 an incomplete plan, it is analyzed by the blackboard controller for cast calls and 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 which Actors get to respond to the new hypothesis once it is posted to the blackboard. Certainly the Actor representing the character of 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 in the narrative, 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.

As Actors progressively decompose each others' abstract steps, at some stage in the generation process an Actor will be unable to avoid planning primitive actions for other characters. This situation is easily handled if the Actor treats that primitive action as if it were abstract, adding it the plan and annotating the hypothesis with a cast call. When the next Actor attempts to refine this hypothesis by decomposing the primitive action, it will be faced with impossibility because the action cannot, in fact, be expanded upon. However, the current Actor cannot be satisfied that an earlier Actor has dictated which primitive action its character will make; idiosyncratic character behavior is most prevalent at the level of primitive actions. Instead, the current Actor removes the offensive primitive action and replace it with another of its choosing. In many cases, the replacement may be trivial: the Actor chooses not to replace the offending action at all. However, we do not guarantee that there is a common set of primitives shared among Actors.

Cast calls are, therefore, the primary mechanism for breaking the monolithic narrative plan into manageable chunks that can be distributed to the various Actor agents. Of course, each agent receives a full copy of the incomplete plan in order to ensure that its planning process is sound and that recombination is unnecessary. The contribution of each Actor agent is the insertion of actions into the narrative plan that both satisfy flaws in the plan structure and also ensure character behavior is handled in an expert fashion.

3.2.2. 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

```
(define (action kill ?char ?victim)
   :precond ((able (kill ?char ?victim)))
   :effect ((not (alive ?victim)))
```

```
(define (action kill ?char ?victim)
   :precond ((able (kill ?char ?victim))
             (believes ?char (evil ?victim)))
   :effect ((not (alive ?victim)))
```

Figure 4. Specification of an action for a generic character (left) and for a moral character (right).

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 deserving of the outcome. One such possibility is shown in figure 4.

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. The use of schemata to specify behavior is both practical and cognitively plausible. Cognitive psychologists believe that humans, through their experiences dealing with the real world, compile large libraries of schemata defining how to behave in various, familiar situations. Decomposition rules also allow for idiosyncrasies to be expressed because abstract actions can be expanded into subsequences that contain actions that are not rationally motivated. There are two mechanisms for inserting actions into a plan: to satisfy the precondition of an action occurring later in the plan and to expand an abstract action. Whereas the former mechanism has an underlying assumption of rationality, the later does not. Character idiosyncrasies are best expressed through actions that seem superfluous [17] and, without customized libraries of decompositions, an Actor would be constrained to only insert actions into the narrative that establish the conditions for other future actions. Furthermore, 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 [19; 13]. 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 can capture preferences about the types of actions the character likes to perform and the elaborateness of the sequence (this is especially pertinent for evaluating the expansion of communication acts where verbosity is a distinguishing trait). The Actor is modifying one instance of the entire plan, so the heuristic is only applied to the actions inserted to fulfill the cast call the Actor is responding to. Note that the customized heuristic functions utilized by Actor agents are not meant to evaluate the "goodness" of the story, but rather the "believability" of the character actions in the small portion of the narrative plan being worked on.

3.3. From plan space to hypothesis space

Actors search for plans within the space of all possible, sound plans [13; 19]. The ACONF system, as a collection of collaborating agents, searches for a complete hypothesis in the space of all possible hypotheses. However, as an Actor searches for an incomplete but sound plan, it necessarily leaves regions of the plan space unexplored; the 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 one 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 very localized portion of the overall problem and what may seem desirable in the local scope may have severe repercussions to the system as a whole; other Actors could be left unable to refine the solution. Since each Actor searches the plan space independently of the others, one can think of each hypothesis as having its own, independent 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 in hypothesis space is shown in Figure 5.

In Figure 5, there are three hypotheses in the hypothesis space on the blackboard: *X*, *Y*, and *Z*. Actor *A*, 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). Once Actor A commits to a plan, hypothesis X is posted to the blackboard with that plan within. 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*, then narrative generation will fail! It is possible that 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.

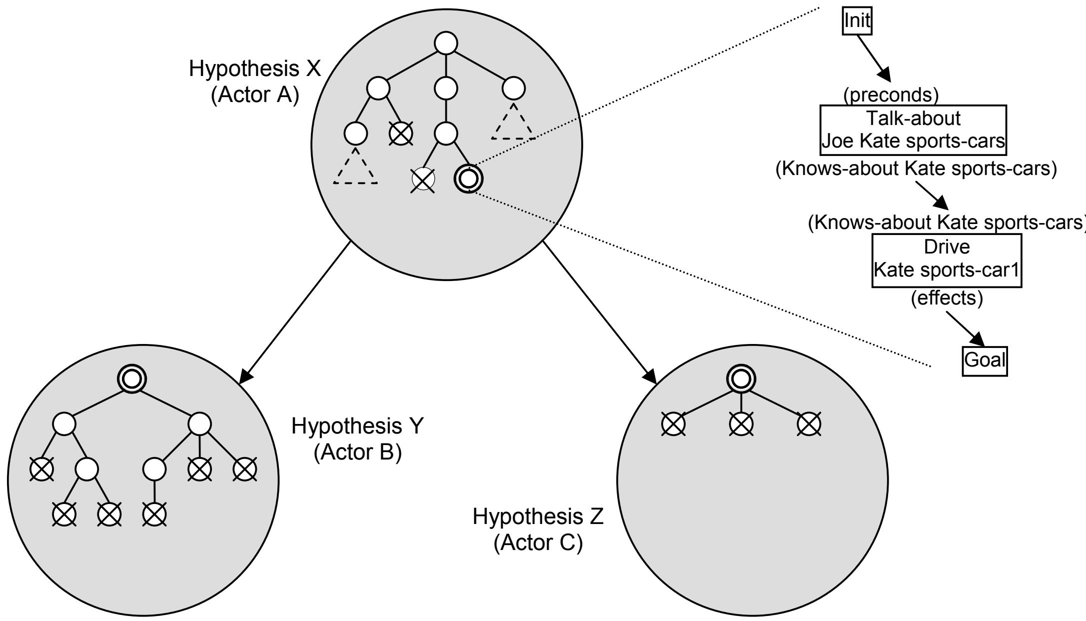

Figure 5. Plan spaces within the hypothesis space. The gray circles are hypotheses. The smaller, embedded, circles are nodes in the narrative plan search space. Each plan space node contains a (possibly flawed) plan, one of which is shown expanded to the right.

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 that 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 5, 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 5 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 [13], 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. Depending on how hypothesis ranking occurs, this leads to the possibility that revisiting a hypothesis could be more favorable than visiting a previously unexplored hypothesis. This is perfectly natural behavior for the ACONF system which considers the possibility that a hypothesis can be improved upon, thus leading to a better overall solution.

With revisitation, the hypothesis search problem in ACONF is as complete as the partial-order planner used by the Actors in the system. Decompositional, partial-order planning itself, however, is not complete [19].

3.4. Author goals

Every Actor uses plan space search heuristics to find a segment of the overall narrative plan that best illustrates character traits. The blackboard controller also uses heuristics functions to choose the best hypotheses to explore or revisit. The heuristic functions used by the blackboard controller are responsible for guiding the generation process to a plan that is not only sound and complete, but is also a good story. The planning algorithm results in plans that have temporal and causal structures that are associated with the temporality and cause-and-effect nature of narrative fabulas [20] and, consequently, have plot coherence. Character believability is handled by the distribution of parts of the planning process to expert agents. As important as plot coherence and character believability is to narrative generation, it is still possible to have both but not have a good story. For good story to be generated, a system requires a model of narrative procession. This is not to be confused with the model of narrative structure; ACONF models narrative structure as a paritally-ordered, hierarchical plan. A model of narrative procession is one that guides the process of generation itself. One such model is Aristotle's dramatic arc [1], show in Figure 6, which makes claims that good stories have a certain pattern of effect on the audience's level of tension. The dramatic arc model is employed in some narrative generation systems such as [10] and [18], which use a battery of domain-specific evalutation functions to measure audience tension. Evaluating a narrative against the dramatic arc requires a complete, totally-ordered sequence of events, something that does not exist in most of the hypotheses in ACONF's blackboard. How to evaluate partial narrative plans for their story quality is an open question.

Lacking a model of narrative progression, ACONF must rely on other forms of guidance to ensure that it arrives at story solutions that are good and not just coherent and believable. ACONF requires the author – a human or other process which understands narrative procession – to provide guidance through a set of author goals. Author goals are temporally ordered descriptions of states of the story world that should appear throughout the narrative but not necessarily at the end. Unlike the plan goal, which is a description of the state of the world the planner must achieve, author goals describe intermediate states of the world. ACONF must achieve each of these intermediate world states. To maintain domain independence, author goals need can be declared at high levels of abstraction, such as "villain character established" or "conflict between protagonist and antagonist." Such high-level goals can be satisfied in many ways, leading to great variety in storytelling. For more fine grained control of the plot, lower level author goals can be specified such as "villain has the Maltese Falcon" or "protagonist is lost in the desert." Author goals are translated into pairs of predicates and roles and inserted into the narrative plan as unresolved goals for the Actors to satisfy. The blackboard identifies Actors whose characters can fill these roles and it is the responsibility of the Actors to generate plan structures that soundly satisfy these goals. Different Actors propose differing hypotheses on how these author goals are to be achieved. For example the author goal, "villain character established" (the role associated with this goal is that of the

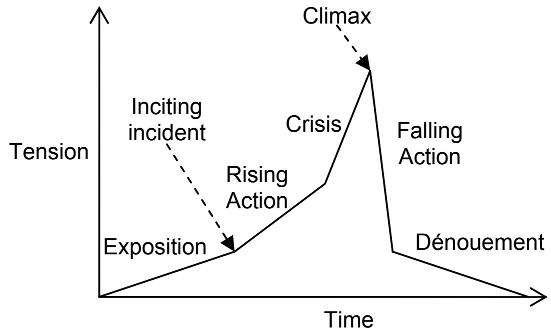

Figure 6. Aristotle's dramatic arc [Aristotle].

villain), is best handled by the Actor of the villain character who might structure this segment of the narrative. The author goal, "conflict between protagonist and antagonist," can be handled by several different Actors. For example, the Actor representing a villain could structure the narrative such that the hero was mortally wounded when walking into a trap laid by the villain (although if there are future author goals specifying the hero's eventual success, this hypothesis will lead to a dead-end). The Actor representing a hero could structure the narrative as a daring escape. Regardless of who responds to the author goals, some sort of conflict must ensue. Thus author goals provide direction to the Actors in the ACONF system on how to structure the narrative without imposing the structure directly on the Actor agents.

Since author goals can be specified at a high-level of abstraction, it suffices to say that Actor libraries must contain decomposition rules capable of rendering high-level author goals into behavioral action sequences. One of the strengths of using expert agents to represent characters during planning instead of using characters directly is that an Actor can reason about presentation. For example, if the author goal is "hero character is established", an Actor can choose the sequences of actions that best portray a back-story for that character.

Since author goals provide intermediate guidance of the narrative structure, they allow for circumstances of reversal of fortune, identified by [15] as an element of good storytelling. A reversal of fortune occurs when some character transitions from a positive circumstance to a negative circumstance or vice versa. Typically good storytelling involves reversals of fortune in which a protagonist transitions from a positive circumstance to a negative circumstance and then back to a positive circumstance. This progression roughly parallels Aristotle's dramatic arc in which reader tension first increases action rises and the protagonist's situation deteriorates and then decreases after the climax and the protagonist's fortunes are reversed. For ACONF to reason about reversal of fortune, it must contain two author goals, P, describing a desirable (from the reader's perspective) world state, and $\neg P$, describing an undesirable world state. The reason it is not inconsistent for ACONF to contain contradictory goals is because the author goals are ordered. First one author goal is achieved and then the next author goal is achieved. The Universe system [8; 9], which similarly uses a decompositional, partial-order planner to generate narrative, cannot handle reversible goals. If the Universe system wants to generate a story about a Dick and Jane who divorce and then later remarry, the story had to be broken into episodes, one of which has the goal "Jane is not married to Dick" and another which had the goal "Jane is married to Dick," each to be run as an independent instantiation.

Author goals are essential to ACONF for initializing the blackboard. A blackboard is essentially an event driven system and events are derived from the hypotheses posted to the shared problem space. For narrative generation to begin, an initial, empty hypothesis is posted to the blackboard containing a list of author goals as annotations.

4. 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. The overall structure of narratives generated by ACONF is guided by author goals, which specify incremental changes in the story world state so that dramatic tension can be manipulated and character fortunes can be reversed.

5. References

1. Aristotle. *Poetics*. (S. H. Butcher, Trans.) Hill and Wang, Publishers: New York 1989.
2. Aylett R. Narrative in virtual environments – towards emergent narrative. In: *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
3. Bailey P. Searching for storiness: story generation from a reader's perspective. In: *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.

4. Bal M. *Narratology: Introduction to the theory of narrative*. University of Toronto Press: Toronto 1997.
5. Bates J. Virtual reality, art, and entertainment. *Presence: The Journal of Teleoperators and Virtual Environments* 1992; 1(1); 133-138.
6. Bruner J. *Acts of Meaning*. Harvard University Press: Cambridge, MA 1990.
7. Carver N, Lesser, V. The evolution of blackboard control architectures. *Technical Report 92-7*. University of Massachusettes: Boston 1992
8. Lebowitz M. Creating characters in a story-telling universe. *Poetics* 1984; 13; 171-194.
9. Lebowitz M. Story-telling as planning and learning. *Poetics* 1985; 14; 483-502.
10. Mateas M.An Oz-centric review of interactive drama and believable agents. *Technical Report CMU-CS-97-156*. School of Computer Science, Carnegie Mellon University: Pittsburgh 1997.
11. Mateas M, Sengers P. Narrative intelligence. In: *Working Notes of the AAAAI Fall Symposium on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
12. Meehan J. Tale-spin, an interactive program that writes stories. In: *Proceedings of the 5th International Joint Conferences on Artificial Intelligence*. 1977.
13. Penberthy J, Weld D. UCPOP: A sound, complete, partial order planner for ADL. In: *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*. 1992.
14. 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.
15. Sgouros N. Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence* 1999; 107; 29-62.
16. Theune M, Faas S, Nijholt A, Heylen D. The virtual storyteller: Story creation by intelligent agents. In: *Proceedings of the First International Conference for Interactive Digital Storytelling and Entertainment*. 2003.
17. Thomas F, Johnson O. *The Illusion of Life: Disney Animation*. Hyperion 1995
18. Weyhrauch P. *Guiding Interactive Fiction*. Ph.D. Dissertation, Carnegie Mellon University: Pittsburgh 1997.
19. 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.
20. Young R. Notes on the Use of Plan Structures in the Creation of Interactive Plot. In: *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*. Mateas M, Sengers P eds. 1999.
21. Young R, Riedl M. Towards an architecture for intelligent control of narrative in interactive virtual worlds. In: *Proceedings of the International Conference on Intelligent User Interfaces*. 2003.