http://liquidnarrative.csc.ncsu.edu/pubs/tr03-000.pdf

NC STATE UNIVERSITY

Liquid Narrative Group
Technical Report Number 03-000

Actor Conference: Character-focused Narrative Planning

Mark O. Riedl
Department of Computer Science
North Carolina State University
Raleigh, NC, 27606
moriedl@eos.ncsu.edu

November 26, 2002

Liquid Narrative Group

Abstract

The ability to generate narrative can be applied to entertainment and educational applications to the benefit of both. In this paper, I informally evaluate several narrative generation systems. Automated narrative generation systems can be classified as character-centric and author-centric techniques. Character-centric systems tend to develop narratives with strong character believability but weak coherent plot lines. Author-centric systems tend to develop narratives with strong plot coherence but weak character believability. I then describe an alternative narrative generation system, the Actor Conference, which benefits from both techniques. The Actor Conference system uses a blackboard architecture to coordinate the efforts of many autonomous agents, each representing a single character in the story world, to generate a single, coherent narrative plan.

1. Introduction

Narrative is an important aspect of everyday life. Over the last decade, researchers have come to realize that narrative is the primary mechanism through which we interpret the experiences of our lives and communicate those experiences with others [6[6]]. This ability to perform narrative thought enables us to make sense of the world around us [6; 17; 21]. Given the important role that narrative plays in human cognition, it is reasonable to assume that the ability to generate narrative could play an important role in human-computer interaction [21], particularly in applications for entertainment and education. Narrative plays an essential part of two entertainment media: movies and computer games. Story is central to the movie viewing experience [6] and is becoming more central to the computer game experience as well. Even in first- and third-person perspective games where simulated violence is the most salient component of game play, story and plot is used to legitimize the goals of the game. As graphics technology becomes more sophisticated and more homogenous, it is often story that becomes the distinguishing feature. Unfortunately, in the computer game industry, where there is ample opportunity for interactivity, the story built into games tends to be a static and unchanging component and thus has more in common with the experience of movie watching than real life. Story tends to be pre-scripted – possibly containing a finite number of branching alternatives – and game play leads to a limited number of outcomes. This alone is motivation for automated narrative generation. Indeed the game-playing experience and game replay value could be greatly enhanced by the ability to generate novel plotlines that tailor themselves to the user.

While entertainment is the most obvious benefactor of the ability to generate narrative automatically, it is important not to overlook education. Since we, as humans, experience the world as narrative, it reasons that we also learn and remember through narrative [6; 19] and that computers can provide constructivist environments for narrative learning [19]. Again, the ability to present tailor narrative environments instead of pre-scripted sequences to the user seems to hold great potential.

1.1. Narrative

While there is ample evidence that narrative is an important part of entertainment and education, it is important to understand what narrative is. Narrative is simply the recounting of one or more real or fictitious events, usually oriented around a single goal, that are related to each other temporally and causally [4; 20]. Even when the cause and effects are not immediately obvious, one is inclined to posit likely causes and effects in order to make sense of the world. Only when events are arranged such that there is no way to attribute causality does one have trouble understanding a series of events as narrative [4; 6]. Stories are prototypical examples of narrative since the events that take place within the story are logically ordered so that cause and effect are apparent. In fact, the terms *story* and *narrative* are often used interchangeably.

Narrative theorists decompose narrative into a hierarchy of components: text, story, and fabula [4]. From the bottom up, *fabula* is the sequence of events that take place in the story world, some of which are exposed to the audience in the text and some of which are hidden. The story world can be the real world in the present time, the real world in historical times, or a fictional world. Within the story world, events that are causally justified take place, although not all events necessarily make it into the text of the story itself. The *story* layer filters the fabula by selecting characters and viewpoint through which to expose parts of the fabula. This process is referred to as focalization. Hence a description of the events in *Beowulf* told from Beowulf's perspective is not the same story as that told from the perspective of the Grendle, even thought the fabula is the same. The final, topmost layer is the *text*. The text is the specific wording and phraseology chosen to tell the story with. The fabula is the essential base from which narrative arises and is the layer at which concepts such as causality exist. When I refer to narrative generation, I am specifically referring to the ability to generate a fabula. I assume some additional processing – such as that done by the AUTHOR system [8] – can be performed to transform the generated fabula into prose.

1.2. Automated narrative generation

Automated narrative generation has been addressed in the past (c.f. [18] and [14; 15]) and in more recent research efforts (c.f. [2], [16; 5], [13], and [22]). Of primary importance here is the tradeoff between plot coherence and issues of character. In general, systems which excel in one aspect of narrative generation tend to be weak in the other. Systems that generate highly coherent narrative structures often neglect issues of character and believability. Systems that capitalize on the use of highly believable characters tend to promote poor narrative structure. In this paper, I 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. Research and development of the ACONF system began in the fall of 2001 at North Carolina State University as a requirement for the author's doctoral qualifying exam. The remainder of the paper is outlined as follows. Section 2 provides a background of related work and a classification framework with which to evaluate existing and future narrative generation systems. This framework also acts to provide motivation for the narrative generation techniques used in the ACONF system. Section 3 provides a detailed description of the narrative generation process used in the ACONF system. Section 4 gives an extended example of how ACONF generates narrative, illuminating some of the strengths and weaknesses of the system. Finally, Section 5 briefly details some of the future work I plan to pursue.

2. A classification framework

In order to understand how ACONF relates to previous attempts at narrative generation, I provide a simple classification framework that classifies narrative generation systems along two continuous dimensions: plot coherence and character believability. *Plot coherence* is defined as how well the events of the narrative relate to achieving a unified experience. A coherent plot can be summarized in relatively few sentences [23], indicating that the narrative is structured around a few concepts. This metric is one shared both by the Disney story team and by Alfred Hitchcock [23]. *Character believability* is the extent to which characters in the narrative exhibit rich personalities, emotion, social behavior, motivations, and personal goals [16]. 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 be divided into groups according to a taxonomy given in [17] (adapted from [3]). According to this taxonomy story-generation systems can be classified as *author-centric*, *story-centric*, or *character-centric* systems. 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 taxonomy of story-generation techniques is tightly coupled with the classification framework. Character-centric systems tend to result in stories with strong character believability but weak plot coherence. Author-centric systems tend to result in stories with strong plot coherence but weak character believability. I do not consider story-centric systems further because they focus solely on grammatical and linguistic constraints of story. The relationship between the taxonomy and the classification framework is expressed in Figure 1.

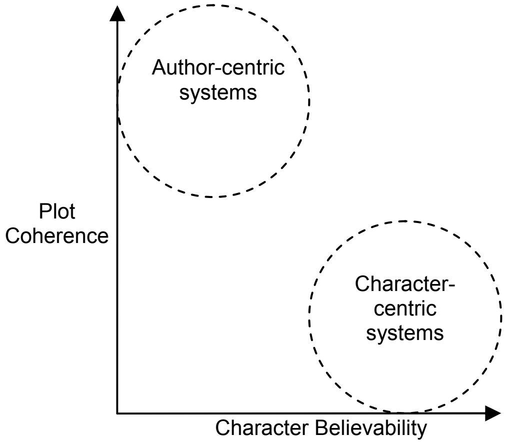

Figure 1. The classification framework for narrative generation systems.

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. Character-centric systems typically situate a user – embodied in a graphical avatar – within a 3D virtual environment populated by autonomous, animated agents. Narrative arises from the interaction with the animated 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.

The OZ project [16] 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 without acting too inconsistently [16; 5].

Klesen et al. [13] developed the Virtual Puppet Theatre in which children were aloud to freely interact with other autonomous agents in a virtual 3D environment. The autonomous agents can perform affective reasoning and behavioral planning, making believable and engaging. The autonomous agents are also cast into roles as protagonists or antagonists and will, accordingly, perform behaviors that either support or confound the goals of the user. The goals of the user are authored in advance based on a simple scenario. For example, the user might be a farmer whose pre-determined goal is to establish order over the farm animals. Some farm animals will behave is ways that increase order while some farm animals will behave in ways that reduce order.

Tale-spin [18] 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 [18]. Unlike other emergent narrative systems, Tale-spin did not incorporate user interaction.

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. Since author-centric systems generate narrative through a structured, rational methodology and are, hence, not plagued by failure in the same way that character-centric systems are. 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 character to take.

The Universe system [14; 15], built by Lebowitz, is an early attempt at building structured narratives. Universe used 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 operated in the soap-opera domain). The sequence of actions chosen by the planner would 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 a fist-fight 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.

Sgouros [22] 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.

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 [14; 15], 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. Planners depend on identifying causal relationships between actions and that those concepts naturally maps to the domain of narrative [27]. A partial-order planner chooses only those actions that are necessary to achieve a goal but as a narrative, the actions in the plan do not necessarily make sense. Characters whose actions are planned by a partial-order planner will perform behaviors that will bring about a certain state of the world, but those actions are not required to be consistent with an 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 "happily ever after" goal. ACONF introduces the possibility for characters to exert their personalities by enforcing the rule that only the character that is to perform an action gets to plan the action, as long as the overall goals of the narrative are still met. Thus characters can be more believable because the actions that the characters choose to perform are more personalized.

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. Each Actor must plan its own actions within the story world and the frame of the plot line and also coordinate with other Actors to ensure a coherent and understandable plot. Instead of a single monolithic system that generates a script for the actors to act out, the system behaves as if the actors confer and agree upon how the narrative should unfold. The system emulates the process of improvisation, in this regard. Improvisation is recognized as the key mechanism in many character-centric story-generation systems (c.f. [16], [13], and [2]).

For a coherent story to unfold, the expert systems – Actors – must coordinate closely with each other. To accomplish this, we use a blackboard architecture. A blackboard is a central control structure through which autonomous agents can share knowledge and work on a common problem. Each autonomous agent is specializes in solving one aspect of the overall problem and can post partial guesses about the final solution, called hypotheses, to the blackboard for other agents to build on. The basic architecture is presented in Figure 2. An overview of blackboard architectures is presented in Section 3.1. A discussion of how the blackboard architecture is used for story-generation is presented in Section 3.2.

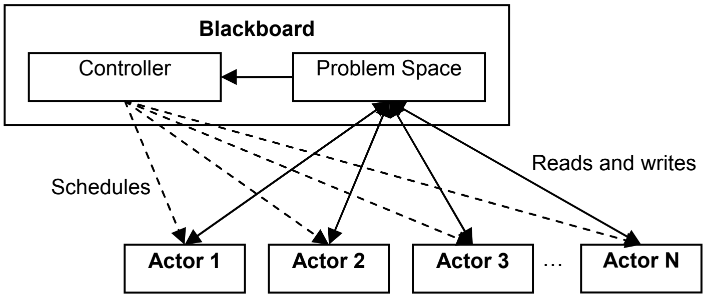

Figure 2. The ACONF system architecture.

3.1. An overview of blackboard systems

The blackboard is a technique for managing collaboration among a large number of autonomous agents that are working together to solve a single problem [8]. 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. 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 the problem towards a solution. One of the strength of the blackboard architecture is the possibility for opportunistic planning [8; 10; 11] which will be beneficial to the problem of generating coherent plotlines with strong character believability.

Blackboard architectures are typically broken up into a global problem space, a controller, and one or more knowledge-sources [8]. The problem space contains many hypotheses – partial solutions – about the solution to the problem. The autonomous agents, referred to as knowledge sources, search the problem space for places in which they can make contributions. If a contribution can be made by a knowledge source, that knowledge source can refine a hypothesis by posting a new hypothesis representing an incrementally more complete solution. The refinement will trigger other knowledge sources to take action. Each hypothesis in the problem space is thus related to all other hypotheses as either a refinement – the hypothesis is an incremental improvement over another hypothesis – or an alternative – the hypothesis is a competing theory about the final solution. Thus the use of a blackboard can be thought of as a search through the space of possible hypotheses. Figure 3 shows a search tree in hypothesis-space. Solid lines indicate a refinement relationship between hypothesis nodes. Dashed lines indicate an alternative relationship between hypothesis nodes (for clarity, Figure 3 does not exhaustively show all alternative relationships). The fringe nodes are hypotheses that have not yet been expanded.

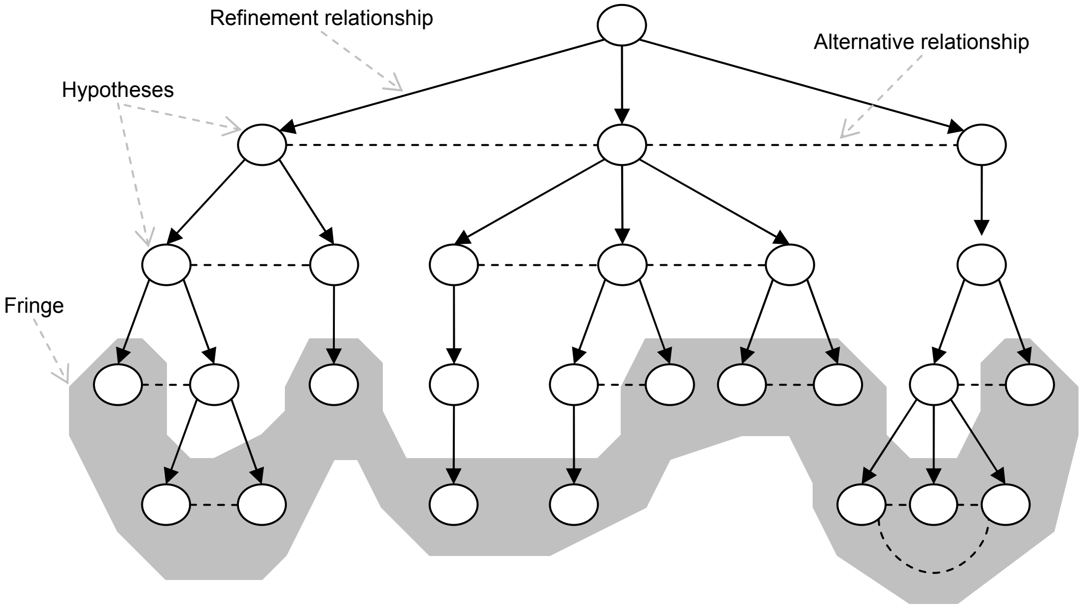

Figure 3. The hypothesis search space of the blackboard.

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 knowledge-sources 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 knowledge sources do not carry over information or experiences when transitioning from one hypothesis to another [8]. 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. A knowledge sources 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.

3.2. Story planning in the Actor Conference system

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 planner [26] which is a decompositional, partial-order planner. When an Actor sees an opportunity to refine an existing hypothesis, it analyzes the incomplete narrative plan contained within the hypothesis for places to expand upon. An incomplete narrative plan has flaws, such as unexpanded abstract actions or steps with unsatisfied goals or preconditions. Some of these flaws can be satisfied by the Actor through planning sequences of actions that that Actor's character can perform. By acting as an expert system for an individual story-world character, a stronger sense of character believability can be brought out in the narrative being generated. The emphasis of an Actor's planning effort is on the actions of its represented character so the plan can remain incomplete if other characters are expected to perform actions; other Actors are responsible for filling in that part of the incomplete narrative plan. Through the blackboard, partial hypotheses are exchanged among Actors until a final, complete narrative is generated.

Any sufficiently interesting story involves character interaction. 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, interfering with the authority of other Actors who are experts on those other characters. 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 other Actors as they refine the plan. The second advantage is that, at a sufficiently high level of abstraction, characters do not exert idiosyncratic behavior. For example, the action, "talk-about Joe Kate sports-cars", captures the essence of narrative at a high level of abstraction without concern that Joe might have a tendency to be long-winded when speaking on the subject of sport-cars. The question arises of what level of abstraction can an Actor no longer plan actions for other characters without hampering other Actor's abilities to exert their character's idiosyncrasies? The answer to this question is handled in the next modification to the planning process. Actors are forbidden from decomposing actions that belong to characters other than the one that that Actor represents. This means that Kate's Actor will not decompose the abstract "speak-about" action even if it was the one who planned it in the first place (perhaps so the precondition that Kate understands sports-cars is satisfied elsewhere in the plan) because it cannot know how Joe will behave while conversing. The narrative plan is thus left with an unresolved flaw: an abstract action remains unexpanded in the plan and a new, incomplete hypothesis is posted to the blackboard, annotated to indicate the opportunity for refinement. Joe's Actor will eventually be given an opportunity to refine this incomplete hypothesis by decomposing the "speak-about" action in a way that best suites the Joe character.

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 the primitive action as if it were abstract, adding it the plan and annotating the hypothesis. When the next Actor attempts to refine this hypothesis by decomposing the primitive action, it will be faced with an 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. Instead, the current Actor can remove the offensive primitive action and replace it with another of its choosing. In many cases, the replacement may be trivial: the Actor choooses not to replace the offending action at all. However, we do not guarantee that there is a common set of primitives shared among Actors.

3.2.1. Actors

What does it mean for an Actor to be an expert on a story world character? Expertise about character must fit into the framework of decompositional, partial-order planning. We capture this expertise in two ways. First, each Actor draws actions from its own private library of planning operators. The Actor's private action library can be thought of as a knowledge-base of how an individual story world character behaves and interacts with the world. Since planning is decompositional, the action library can potentially have any number of abstract actions that the character is capable of performing. Each of these abstract actions can have any number of decomposition rules, specifying different ways to the character can accomplish abstract actions under differing situations. 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. Decomposition rules also allow for idiosyncrasies to be expressed because abstract actions do not need to be decomposed into the most rational action subsequence. This overcomes one of the primary limitations of the use of partial-order planners for expressing character believability: planners insert actions into the plan that achieve the goals and sub-goals and will overlook any action that is not relevant to those goals. Character idiosyncrasies are best expressed through actions that seem superfluous [23]. 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 planner to insert the correct sequence of primitive actions into the plan. These primitive actions can be defined in ways that are unique to that character, e.g. have differing preconditions or effects.

Besides the use of decomposition rules, character expertise is also captured by the use of plan-space search heuristics. The Longbow planner uses search heuristics to a perform best-first search through the plan space. Planning heuristics are typically used to exert domain specific knowledge on the planning process. For instance, a heuristic can favor plans with certain qualities, making the search through plan-space more efficient [26; 24]. Similarly, a heuristic in the hands of an Actor can exert character-specific preferences on the planning process such as a tendency for more elaborate plans or for plans that use certain actions [26].

3.2.2. Cast calls

When an Actor refines a hypothesis, it potentially leaves abstract actions that are unexpanded. This occurs when it needs to plan actions for characters other than its own in order to satisfy its own character's actions. For example, suppose Kate is a character that knows nothing about sports-cars but needs to drive one. Kate's Actor inserts the action, "drive Kate sports-car1", which has many preconditions, one of which is that Kate knows how to drive a sports car. In order to satisfy this precondition, Kate's Actor might also insert the action, "talk-about Joe Kate sports-cars", into the plan so that the character Joe will tell Kate enough about sports-cars for her to drive one. Since "talk-about" is an abstract operator and belongs to a character other than Kate, it will, therefore, be left unexpanded. The hypothesis built by Kate's Actor which includes the "drive" and "talk-about" actions is incomplete when posted to the blackboard. But before the hypothesis is posted, Kate's Actor must annotate the plan in some way to indicate where the flaws are and which Actors, if any, are best suited to resolve those flaws.

Naturally, the annotations could indicate the Joe's Actor should be the one to refine the new, incomplete hypothesis. However, Kate's Actor's motivation is only that some character talks to the Kate character about sports cars so that the narrative plan is sound when it is completed. Any character that is knowledgeable about sports cars could take the place of Joe and therefore, other Actors besides Joe's Actor could propose alternative hypotheses. Thus the annotations Kate's Actor makes to the new, incomplete hypothesis specify a role instead of a specific character's name. Thus this special type of annotation is referred to a cast call because it calls upon any number of Actors to step forward and participate in the narrative. Any Actors whose character matches this role can refine this new hypothesis and inserts its character's presence into the narrative. The Actor must simply replace any bindings to the Joe character with bindings to its own character and resolve the narrative plan's flaw. The hypothesis posted to the blackboard by Kate's Actor can potentially have many refinements, one for each Actor that identifies with the role specified by Kate's Actor in the hypothesis's annotations. Each of these refinement hypotheses will potentially have cast calls which will lead to many further refinements. The process continues until a complete narrative plan is found.

3.2.3. From plan spaces to hypothesis spaces

Actors search for plans within the space of all possible, sound plans [26; 24]. 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. However, there may be many possible candidate plans that the Actor could 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 valid in the local scope may have severe repercussions to the system as a whole; other Actors could be left unable to refine the solution. However, since each Actor searches the plan space independently of the others, 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 4.

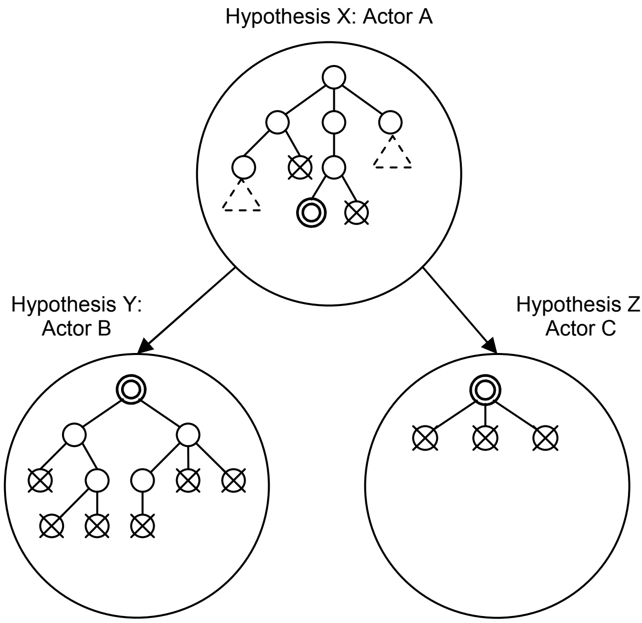

Figure 4. Plan spaces within the hypothesis space.

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 portions of the plan space. The plan space is shown as the tree structure inside the hypothesis. Each smaller circle is a plan in the plan space. Circles that are crossed out represent plans that led to dead-ends. The dashed triangles represent branches of the plan space that have not been explored. The double-lined circle represents the sound plan that Actor A commits to. 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 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 hypothesis – part of a different sub-problem that only Actor A was qualified to solve. Because the hypothesis space is unrelated to plan space, we are threatened by the possibility that narrative generation in ACONF is incomplete.

This level of incompleteness is unacceptable. Therefore, we have modified the blackboard controller to allow hypotheses to be revisited. Revisitation is different than backtracking. Backtracking in hypothesis space means to choose an alternative hypothesis to expand. For example, when hypothesis Y fails in Figure 4, ACONF backtracks to try an alternative, hypothesis Z (which also fails). However, it is clear that the only hope of finding a complete narrative in the example in Figure 4 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 [24], 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. The trick is to define hypothesis ranking algorithms that avoid thrashing – consistently revisiting the same hypothesis when other unexplored hypotheses exist. After all, the plan space in any given hypothesis can be infinite.

With revisitation, the ACONF system is as complete as the partial-order planner used by the Actors in the system. Decompositional, partial-order planning itself, however, is not complete [25; 26].

3.3. Author goals

As mentioned previously, we computationally model narrative as a partially ordered plan. This representation provides us structures that mimic the temporality and causal-and-effect nature of narratives [4; 27] and also provides us with properties such as soundness that we can use to evaluate the coherence of plot. By adopting this representation, we can perform rational computation on the narrative itself during construction of the narrative. The algorithms for partial order planning, however, should not be confused with the model of narrative generation. The model of narrative generation is the process of deciding the overall structure of the narrative which sits above and guides the planning process used by the Actor agents. One such model is Aristotle's dramatic arc [1], shown in Figure 5, which describes the change in the tension felt by the audience over time. The dramatic arc model is used in systems such as [16]. The dramatic arc model as described by Aristotle is, unfortunately, not computational. Lacking a computational model of narrative generation, the ACONF system does not use any model of narrative generation. Instead, the ACONF system relies on *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.

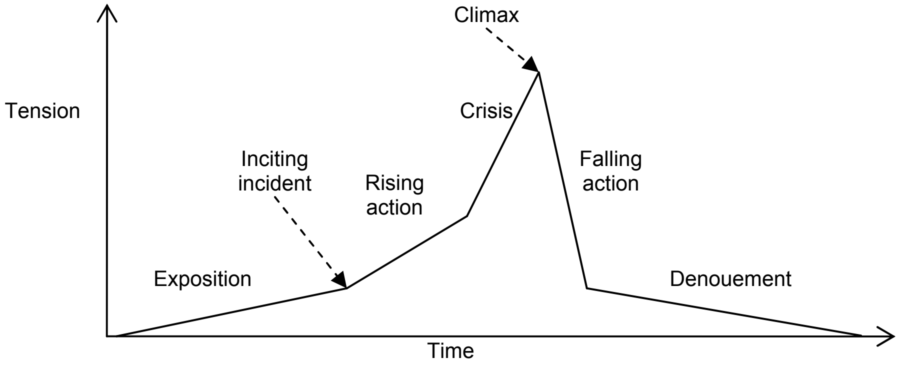

Figure 5. Aristotle's dramatic arc.

Without a model of narrative generation, author goals must be provided to ACONF by an author – a human user or another system who specifies the parameters for narrative generation. However, author goals need only describe states of the story world 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 will identify 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 can 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 villain), would be 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," could probably be handled by several different Actors. The villain's Actor 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 hero's Actor 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.

The use of author goals in substitution for a more comprehensive model of narrative generation is less than ideal. First of all, in order of Actors to handle author goals, the Actor action libraries must contain action operators that have author goals as effects. This can best be achieved by providing abstract operators that satisfy author goals and by providing decomposition rules for these operators that lay out sequences of character actions or other abstract operators that further refine the structure of the narrative. Providing operators for achieving author goals is a natural extension to the existing facilities that Actors already possess for refining hypotheses. It makes sense for Actors to have these operators since Actors are not the actual characters themselves and, thus, should be allowed to reason about the structure of the narrative that best suites their characters. However, the Actor action libraries must now contain operators for both actions within the narrative and for structuring the narrative, which is perhaps an unintuitive use of action libraries.

Author goals are not lists of predicates describing the state of the story world at the end of the narrative plan. Instead, each author goal describes the state of the story world after a subsequence of actions. This is important because author goals can be *reversed*, meaning that if one author goal contains the world state predicate *P*, then another author goal can contain the world state predicate, $\neg P$ without resulting in an inconsistent plan. The Universe system [14; 15], which similarly uses a partial-order planner to generate narrative, could not handle reversible goals. If the Universe system wanted 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 had the goal "not together Jane Dick" and another which had the goal "together Jane 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 hypothesis is posted to the blackboard containing a list of author goals and an empty plan. The author goals are listed as a type of annotation in the hypothesis.

4. An extended example

In this section we provide an extended example of the narrative generation process in the ACONF system. The example is based around the legend of Robin Hood, although any resemblance to historical fact is entirely coincidental. We start off with a multitude of possible characters in the story world: King John, the Sheriff of Nottingham, a Black Knight, Robin Hood, Friar Tuck, and a multitude of peasants. Each character is represented by a uniquely defined Actor agent. As shown in Figure 6a, the blackboard initially contains a single hypothesis, *H0*, containing an empty plan and a set of author goals: *ACHIEVE1* and *ACHIEVE2*. ACHIEVE1 states that a goal of the system is to "establish-character ?x" where ?x will be bound to any character who fits the "villain" role. Similarly, *ACHIEVE2* states that a goal of the system is to establish the hero character.

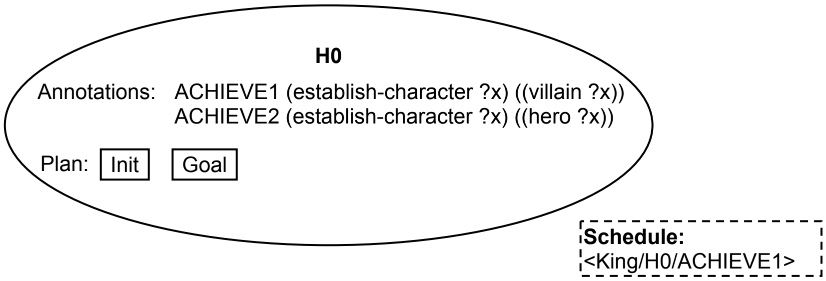

Figure 6a. The initial hypothesis.

The blackboard controller examines hypothesis *H0* for one thing that can trigger the knowledge sources for refinement. It arbitrarily finds *ACHIEVE1* and determines that the King's Actor can respond to this trigger since the King is the only character defined to be a villain in the story world. The King's Actor is scheduled to refine hypothesis *H0* with trigger *ACHIEVE1*. The King's Actor executes, creating a new hypothesis, *H1*, which is a refinement of *H0*. The blackboard is updated as in Figure 6b. In the new hypothesis, the plan is modified as follows. Action 1 is inserted because its effects satisfy *ACHIEVE1*. Furthermore, since Action 1 is abstract, it decomposes the action into Actions 2 and 3 that together state that the King will order taxes and the Sheriff will collect the taxes; taxation is something that kings do by ordering through their henchmen. Action 2 is decomposed because it is an abstract action that the King character performs. Action 3 is not decomposed because it belongs to another character. Since the King' Actor has left a flaw, *H1* is annotated to reflect the fact that Action 3 should be decomposed by any character with a fitting role. The role is determined by analyzing the preconditions of the action.

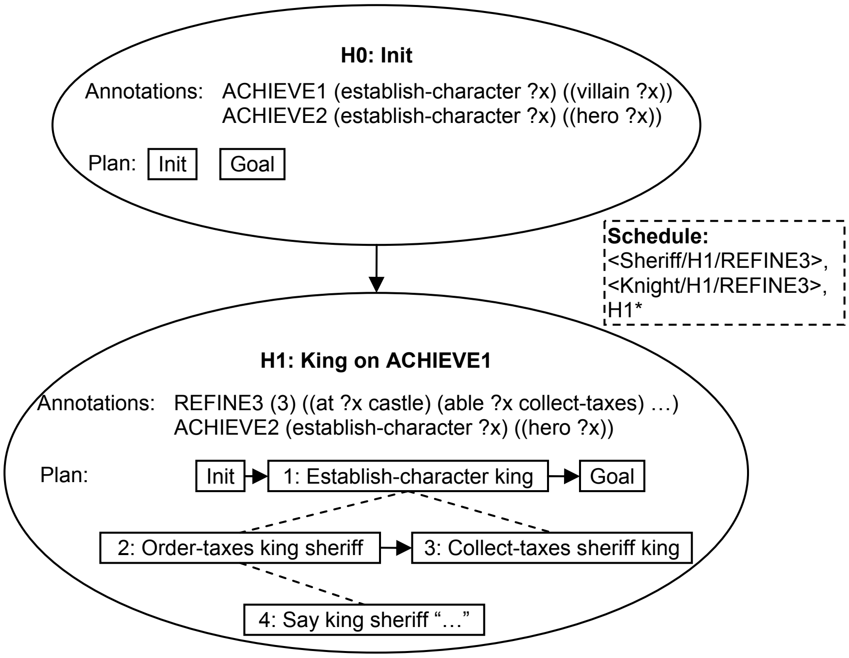

Figure 6b. The refinement of the initial hypothesis.

Hypothesis *H1* suggests that the Sheriff character should be the one to collect taxes, although there are other characters that fit the role equally well. The blackboard controller examines the newly posted hypothesis, *H1*, and arbitrarily picks *REFINE3* as the trigger. The blackboard controller determines that both the Sheriff and the Knight fit the role requirements and each character's Actor is scheduled to refine *H1* on the *REFINE3* trigger. Note that hypothesis H1 is left in the schedule queue so that it can be revisited later, if needed. The asterisk indicates the number of times that hypothesis has been visited. The schedule queue is sorted according to heuristics that rank the likelihood of success. In most cases, unvisited hypotheses will be scheduled earlier than visited hypotheses, although there is no hard rule enforcing this.

First in the schedule shown in Figure 6b is the Sheriff's Actor, which creates hypothesis *H2*, decomposing Action 3 as shown in Figure 6c. All the new actions are primitive but Action 7 belongs to a character other than the Sheriff and is treated as if it were abstract and unexpanded. As before, the new hypothesis is examined and the blackboard controller schedules two peasants, Peasant1 and Peasant2, to refine *H2* on trigger *REFINE4*. However, the next scheduled event is for the Knight's Actor to have a shot at *H1*.

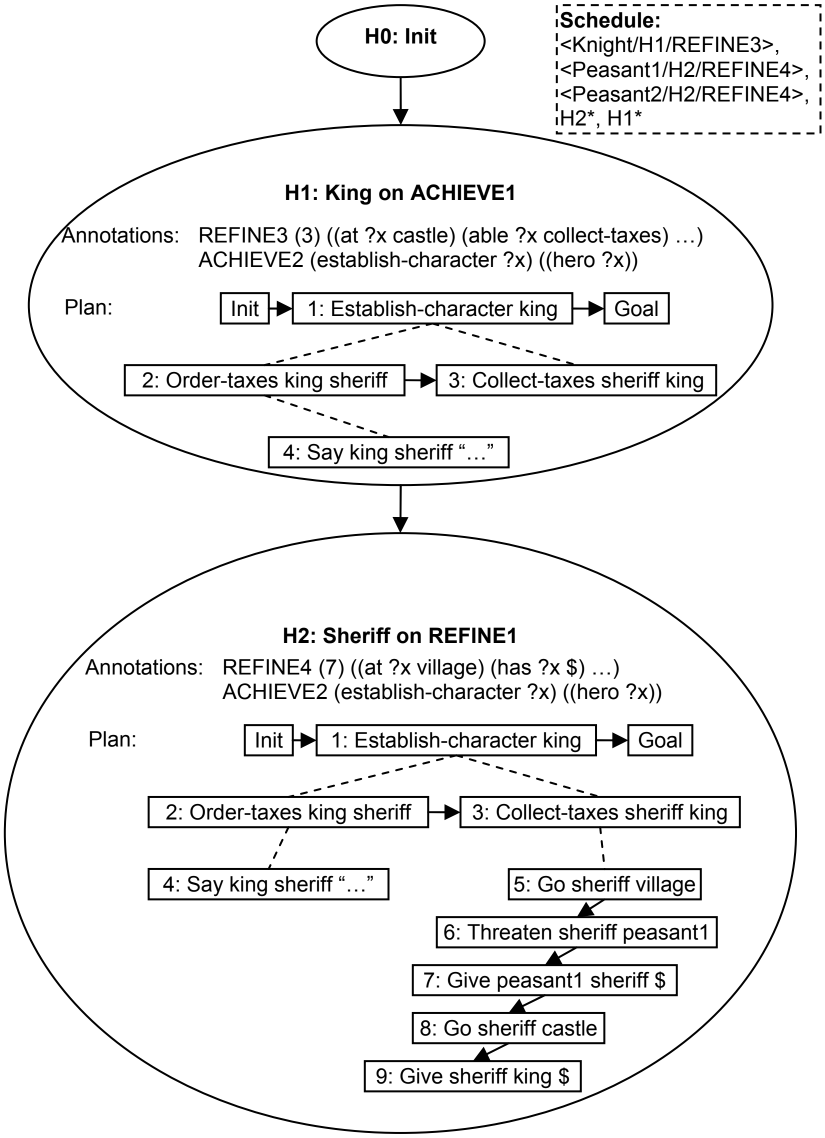

Figure 6c. One alternative for decomposing an abstract action.

The Knight's Actor creates hypothesis *H3* to decompose Action 3. The Knight's Actor's action library which differs from that of the Sheriff's Actor, happens to have two decomposition rules corresponding to the "Collect-taxes" abstract action. Of course, the Actor cannot more than one solution at a time, so the most promising decomposition, shown in Figure 6d, is used. The plan space in hypothesis *H3* will have at least one unexplored branch in which the alternative decomposition rule is pursued, which can be explored if *H3* is ever revisited. As shown in Figure 6d, this time no actions belonging to characters other than the Knight have been added so there are no new annotations for the blackboard controller to consider.

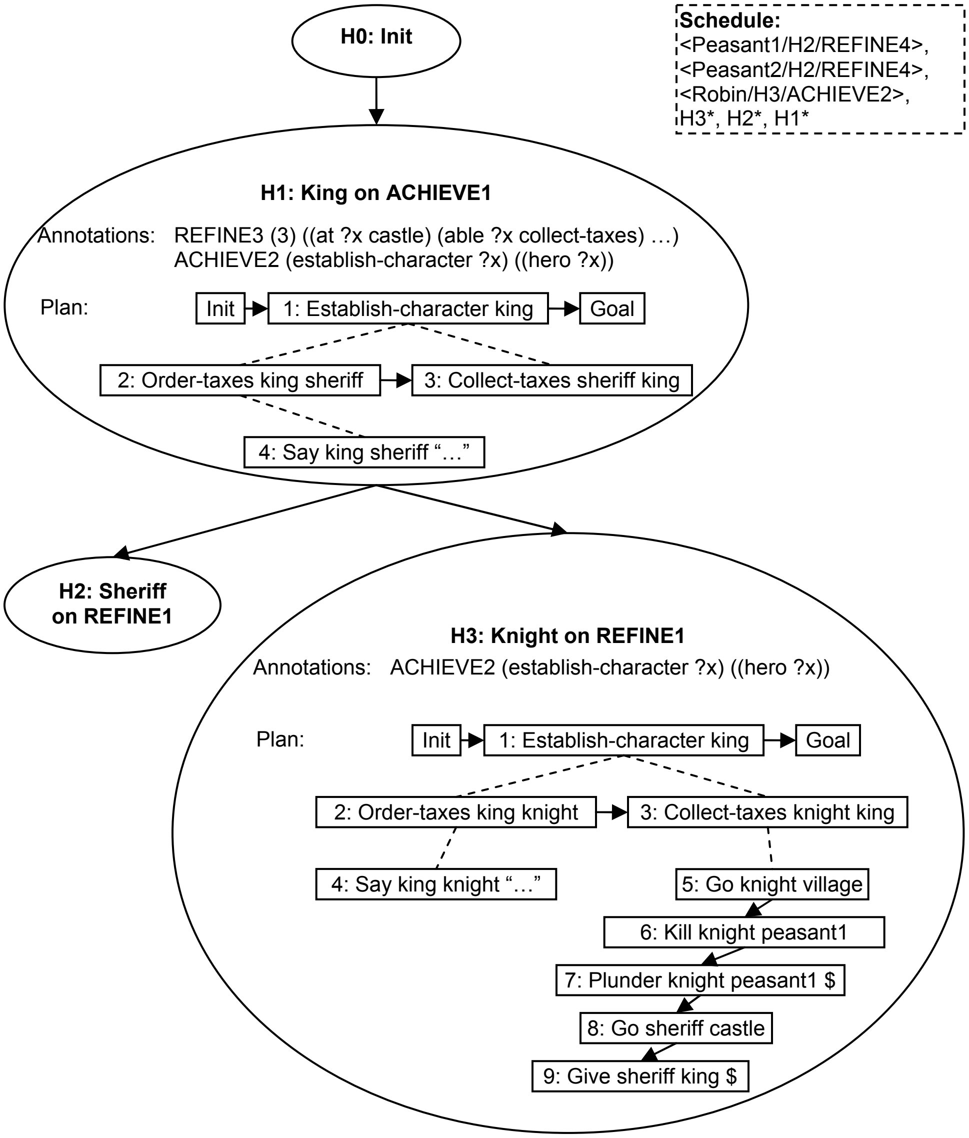

Figure 6d. A second alternative for decomposing an abstract action.

Next on the blackboard controller's schedule is Peasant1's Actor, which will attempt to refine hypothesis *H2* on the *REFINE4* trigger. As mentioned before, the REFINE4 annotation suggests that Action 7, as shown in Figure 6e, is abstract and should be decomposed. Peasant1's Actor will quickly realize that Action 7 is in fact primitive and inserted by an Actor that does not have authority over the Peasant1 character. Action 7 will be removed, leaving open preconditions on Action 8. The planner is invoked to resolve these open preconditions from actions in Peasant1's Actor's action library. In most cases, planning will terminate quickly, replacing the missing action with one that is very similar. The new "Give" action that is inserted, since it comes from a different action library, may be defined to have different preconditions and additional work must be performed by the planner to resolve any additional flaws or threats. For the purpose of the example, the new "Give" action does not differ significantly from the original "Give" action and, as a result no additional annotations are created. Peasant2's Actor's go at hypothesis H2 ends up in similar results (maybe Peasant1 and Peasant2 use a default action library, since it is not economical to develop elaborate Actors for characters who will have minor roles in the narrative). Figure 6e shows the state of the blackboard after Peasant1 and Peasant2 are through.

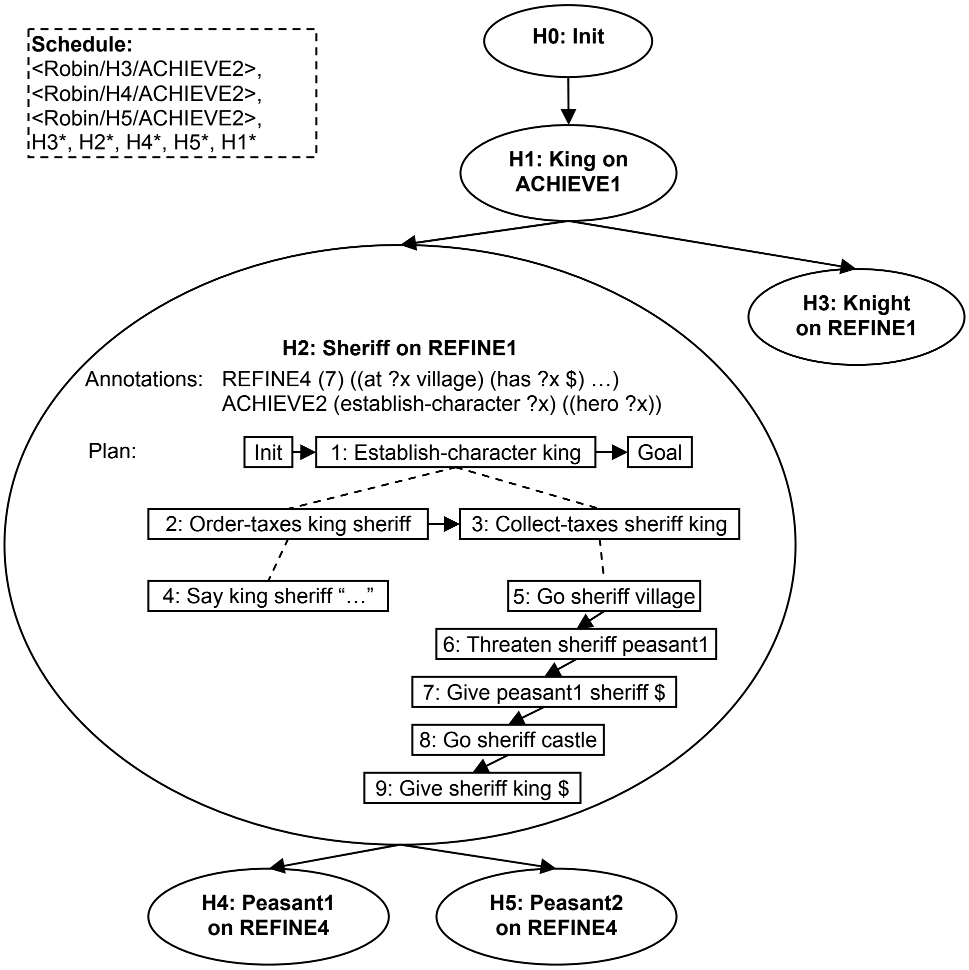

Figure 6e. Further search through the hypothesis space.

At this point in the example, *ACHIEVE2* becomes the salient trigger. The fact that *ACHIEVE2* has not been considered before now is a result of the arbitrary order in which the blackboard controller finds triggers in the hypotheses on the blackboard. The order that triggers are discovered should not affect the ability of ACONF to generate narratives. The Actor corresponding to the character, Robin, is the next to be scheduled, attempting to refine hypothesis H3. Suppose that Robin's Actor fails to find a plan that satisfies the author goal to establish the hero character. This could happen as a direct result of the death of the peasant – perhaps the plan was to steal the money back from the King and return it to the rightful owner. Due to the failure to find a plan, no new hypothesis is returned to the blackboard and the blackboard controller continues planning from the unmodified schedule queue.

Suppose also that the next two refinements also fail, although there is no evident reason why this would be so. The resulting blackboard configuration is shown in Figure 6f. The schedule queue now only contains hypotheses that have already been visited at least once; revisitation occurs.

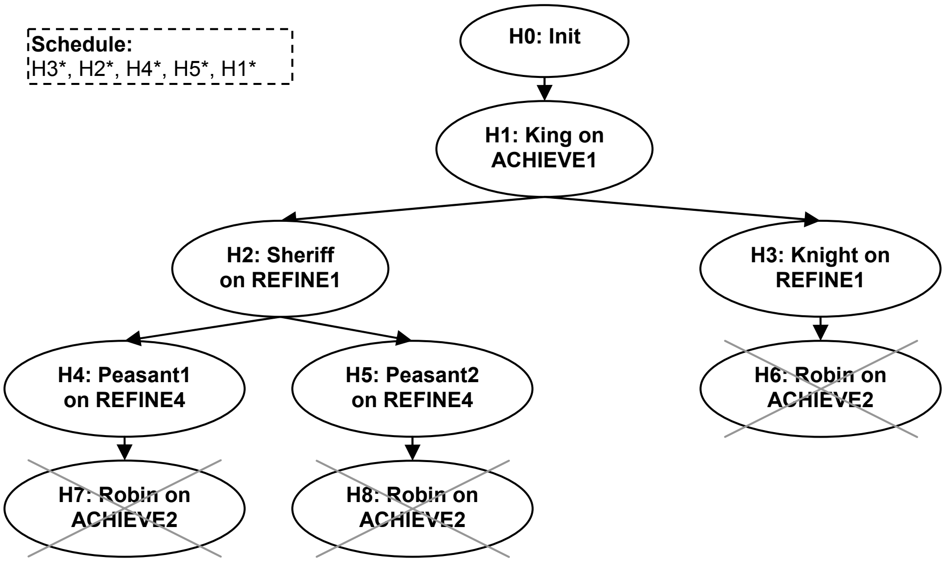

Figure 6f. The state of the blackboard requiring revisitation to occur.

The first hypothesis to be revisited is hypothesis *H3*. Recall that the Knight's Actor had a choice of two decomposition rules when *H3* was originally created. When *H3* is revisited, unexplored branches of its associated plan space – one of which corresponds to the use of the alternative decomposition rule – are iteratively explored until a second plan, if any, is found. The new plan generated by *H3* is shown in Figure 6g. The differences between the new plan and the original plan, shown in Figure 6d are highlighted. Note also that the new plan presents opportunities for further refinements because it includes a primitive action that does not belong to the Knight character. A new annotation is created marking this opportunity and H3 is updated on the blackboard. The blackboard controller does not distinguish between new hypotheses posted to the blackboard and old hypotheses that are updated, so the blackboard controller analyses the updated hypothesis for opportunities for refinement and schedules Peasant2's Actor (apparently Peasant1 is not a loyal-citizen and is not selected as a candidate). As before, *H3* is not removed from the schedule queue, but is re-ranked and reinserted as shown in Figure 6g.

The process of scheduling Actors and refining hypotheses continues as normal from the revisited hypothesis, *H3*. Supposing that Peasant2's refinement of hypothesis *H3* resolves in a way similar to hypotheses *H4* and *H5*, the resulting state of the blackboard is shown in Figure 6h. Narrative generation could even come to a successful conclusion if the remaining author goal, *ACHIEVE2*, is satisfied without introducing any new flaws to the hypothesized narrative plan.

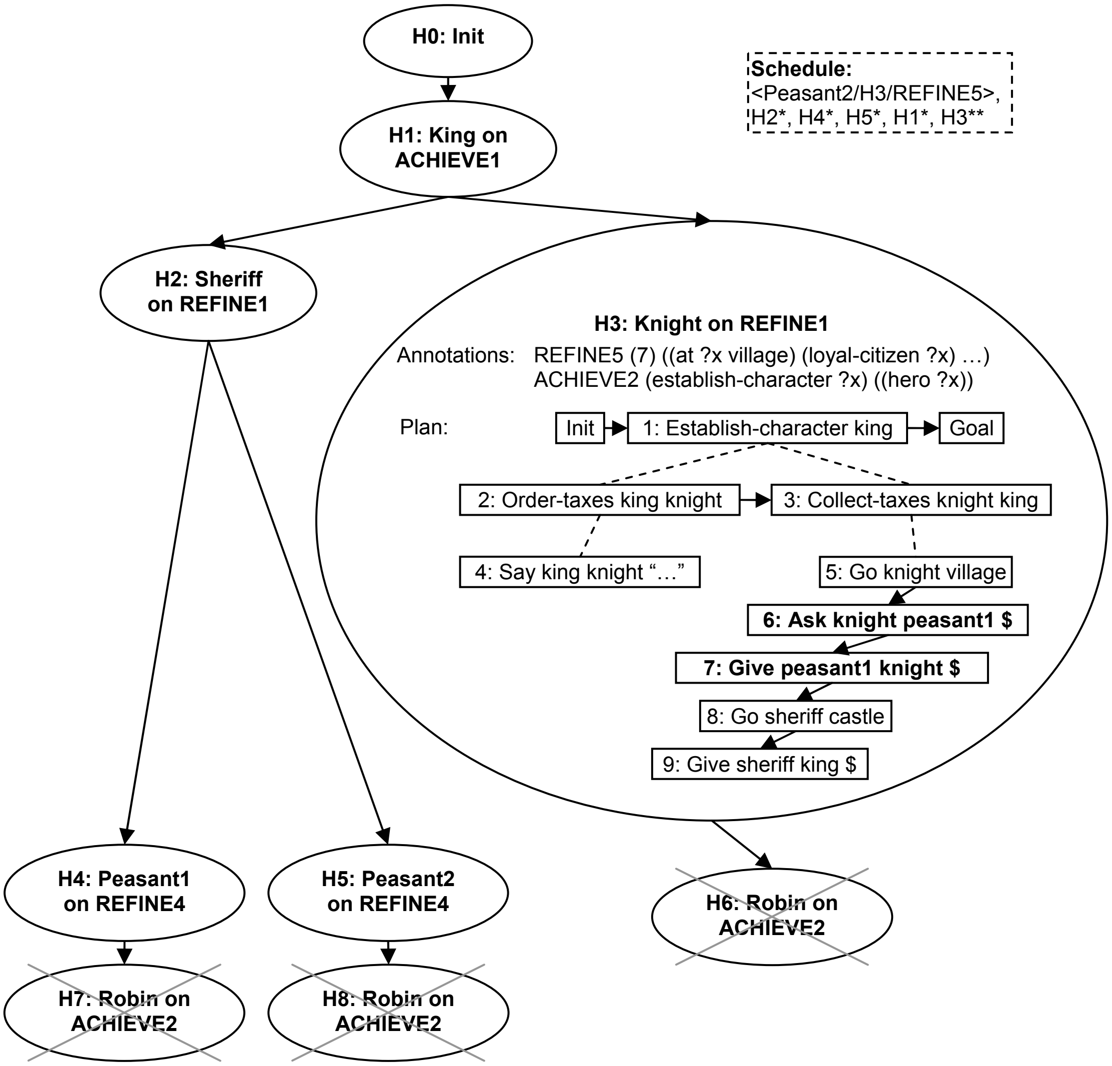

Figure 6g. The revisitation of hypothesis *H3*.

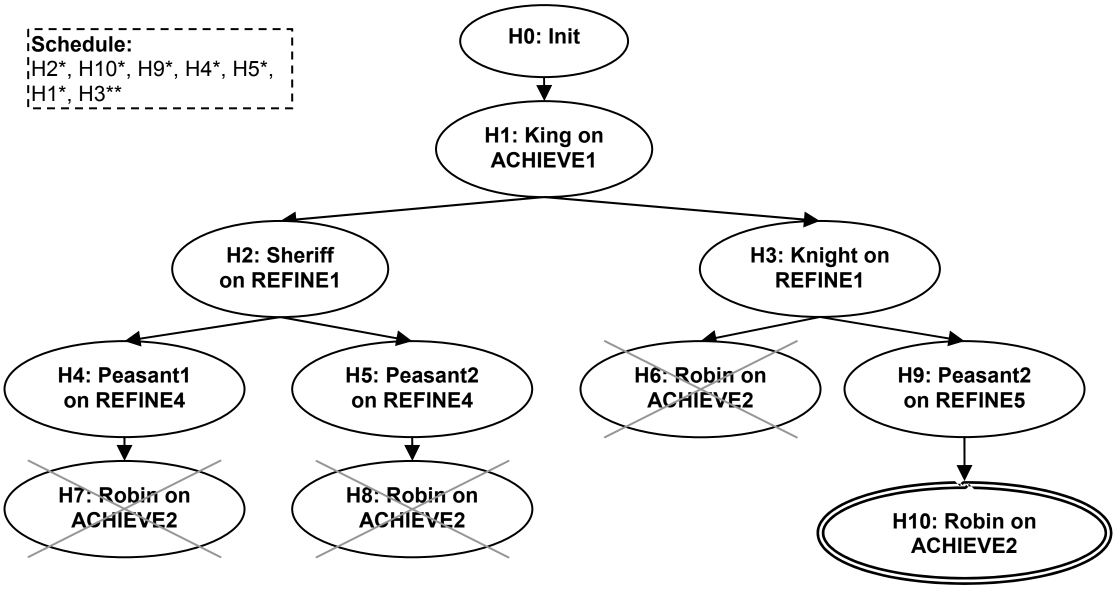

Figure 6h. The final state of the blackboard.

5. Complexity analysis

The Actor Conference (ACONF) system uses a modified version of the Longbow Planner [25; 26]. The Longbow planner is a decompositional, partial order planner that extends the UCPOP algorithm [24] to include hierarchical decomposition. The complexity of UCPOP is $O(cb^n)$ where the three parameters are as follows.

- *n* is the number of open conditions that must be addressed. One can also think of this as the number of nondeterministic choices the planning algorithm must make.

- *b* is the average branching factor of the plan space. One can think of this as the average number of possibilities to be considered at each nondeterministic choice point.

- *c* is the cost of processing a single node in the plan space. The cost is dependent on the size of the node, which varies. The cost, however, does not significantly impact the complexity of the planning algorithm [24].

The Longbow planner, by incorporating hierarchical decomposition into the planning framework, appears to increase the complexity of the planning algorithm. At every nondeterministic choice point, instead of merely choosing open conditions to resolve, the Longbow algorithm may also choose an abstract step to expand. This additional choice increases the size of *n*. However, in practice, we find that hierarchical decomposition often reduces the difficulty of planning because chunks of the plan can be inserted based on predefined decompositional schema.

The Longbow planning algorithm, however, is modified in ACONF so that decomposition is sensitive to the Actor agent that is running the planning algorithm. Thus if an Actor plans an abstract step that refers to another Actor, then that step is not decomposed. Thus not all abstract steps are decomposed, so the value of n is smaller than the value of *n* for the unmodified Longbow algorithm. The value of *n* for the Actor-sensitive Longbow algorithm is still larger than that of UCPOP because some decomposition still takes place. We refer to this value as *n**.

The individual planning efforts of any individual Actor is thus computationally less complex than building a complete narrative plan in a monolithic process. This is obvious observation because parts of the plan will be reserved for other Actors. However, since we coordinating the individual planning efforts of each Actor within the context of a larger hypothesis search space, we must also consider the complexity of searching the hypothesis space. We do this by defining three more parameters, *a* and *m*, as follows.

- *a* is average number of Actors that can respond to any given cast-call. More formally, *a* also corresponds to the average branching factor of the hypothesis space. In most cases, the branching factor will be low because the role specified by the cast-call will often be quite specific, excluding most Actors from revising a hypothesis. However, in the case where a hypothesis is revisited, any single Actor can revise that hypothesis more than once since each visitation will result in different cast-calls being generated.

- *m* is the number of nondeterministic choices made about which Actor will revise a given hypothesis. This value corresponds to the number of cast-calls generated during narrative planning, including cast-calls generated during hypothesis revisitation.

Taking these factors into consideration, the algorithmic complexity of ACONF is $\mathrm{O}\left(ca^m b^{n^*}\right) \in \mathrm{O}\left(a^m b^{n^*}\right)$. Contrary to what one might expect, the ability to revisit hypotheses does not result in increased computational complexity. Rather it is the ability for multiple Actors to respond to the same cast-call that increases the complexity. Revisitation corresponds to filling out a portion of the overall plan space that was pruned by the modified Longbow algorithm. Thus if, only one Actor ever responds to any given cast-call, then $a^m = \frac{b^n}{b^{n^*}}$; the complexity of ACONF is $\mathrm{O}\left(ca^m b^{n^*}\right) \in \mathrm{O}\left(cb^n\right) \in \mathrm{O}\left(b^n\right)$, which is no worse than partial-order planning in general. The intuition behind this is as follows. Think of the narrative planning problem as having one single plan space instead of many sub-spaces. For every abstract step in the plan that is not decomposed in the modified Longbow algorithm, part of the overall plan space is pruned. However, for every abstract step that is not decomposed, a cast-call is created resulting in the pruned branches of the overall plan space being searched. If there is only one Actor that ever responds to a single cast-call, then $a^m b^{n^*} = b^n$ because the exact same plan space will be searched. However, it is through the ability for many Actors to respond to the same cast-call that provides the opportunity for richness of character believability, without which the development of ACONF would be pointless. Regardless, the special case is important to note because Actors are autonomous agents coordinated by a blackboard. This means that each Actor can operate simultaneously if distributed computing is utilized. Thus, despite the fact that more than one Actor can respond to the same cast-call, ACONF will perform as if there were only one Actor responding; the complexity is no worse than partial-order planning.

6. Future work

Future work on the ACONF system falls in three arenas. The first arena involves the development of a computational model of narrative generation that will complement the multi-agent blackboard approach to narrative planning. It was mentioned previously that allowing Actors to hypothesize about the structure of the narrative, while clearly feasible, is not ideal. While we do not have a clear idea of what this computational model of narrative generation will be, we envision one or more special knowledge sources, similar to the Actors that currently exist, which will encapsulate the logic for structuring the narrative. This computational model of narrative generation will enable us to get rid of the concept of author goals, further minimizing the amount of specification and intervention a human author must exert over the system.

The second arena of future work involves investigating heuristics for best-first search through hypothesis space. Any hypothesis ranking algorithm will be responsible for evaluating the “storiness” [3] of a hypothesis – how likely a hypothesis will lead to a final, acceptable narrative. Determining the suitability of a hypothesis depends on having a computational understanding of what makes a good narrative. Such a model of what makes a good story will have to be mapped into the structures and terminology of partially ordered, decompositional plans.

A final arena of future work involves exploring new ways of taking advantage of opportunism in the blackboard planning process. The ability to plan opportunistically with blackboard architectures has been frequently noted (e.g. [8], [10], and [11]) and one way to take advantage of opportunism is through the use of annotations. A knowledge source – an Actor in this case – can annotate a hypothesis in any number of ways. The annotations signaling flaws in the narrative plan is just the most basic, and necessary, use of this mechanism. Other possibilities include annotating the hypothesis for opportunities for *reincorporation*. Reincorporation is a term used by Johnstone [12] to indicate the way in which entertaining narratives reuse the same characters and concepts throughout. Johnstone does not describe how to use the concept of reincorporation successfully, but is convinced that it is one of the things that make for good narrative.

7. References

[1] Aristotle. (350 B.C.E.). *Poetics*. (S. H. Butcher, Trans.) Available: http://classics.mit.edu/Aristotle/poetics.html.

[2] Aylett, R. (1999). Narrative in virtual environments – towards emergent narrative. In M. Mateas & P. Sengers (eds.) *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, 1999.

[3] Bailey, P. (1999). Searching for storiness: story generation from a reader's perspective. In M. Mateas & P. Sengers (eds.) *Working Notes of the AAAAI Fall Symposium on Narrative Intelligence,* 1999.

[4] Bal, M. (1997). *Narratology: Introduction to the theory of narrative*. Toronto: University of Toronto Press.

[5] Bates, J. (1992). Virtual reality, art, and entertainment. *Presence: The Journal of Teleoperators and Virtual Environments*, 1(1): 133-138.

[6] Baumeister, R.F. & Newman, L.S. (1994). How stories make sense of personal experiences: motives that shape autobiographical narratives. *Personality and Social Psychology Bulletin*, 20(6), 676-690.

[7] Branigan, E. (1992). *Narrative Comprehension and Film*. New York: Routledge.

[8] Callaway, C.B. (2000). Narrative prose generation. Doctoral Dissertation, North Carolina State University Dept. of Comp. Sci.

[9] Carver, N. & Lesser, V. (1992). The evolution of blackboard control architectures. *Technical Report 92-7*, Department of Computer Science, University of Massachusettes.

[10] Erman, L.D., Hayes-Roth, F., Lesser, V.R., & Reddy, D.R. (1980). The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12, 213-253.

[11] Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, 26, 251-321.

[12] Johnstone, K. (1981). *Impro: Improvisation for the Theatre*. Routledge.

[13] Klesen, M., Szatkowski, J., & Lehmann, N. (2001). A dramatized actant model for interactive improvisational plays. In A. de Antonio, R. Aylett, & D. Ballin (eds.) *Proceedings of the 3rd International Workshop on Intelligent Virtual Agents, Lecture Notes in Artificial Intelligence 2190*. Heidelberg: Springer-Verlag. Pp. 181-194.

[14] Lebowitz, M. (1984). Creating characters in a story-telling universe. *Poetics*, 13, 171-194.

[15] Lebowitz, M. (1985). Story-telling as planning and learning. *Poetics*, 14, 483-502.

[16] Mateas, M. (1997). An Oz-centric review of interactive drama and believable agents. *Technical Report CMU-CS-97-156*, School of Computer Science. Pittsburgh, PA: Carnegie Mellon University.

[17] Mateas, M. & Sengers, P. (1999). Narrative intelligence. In M. Mateas & P. Sengers (eds.) *Working Notes of the AAAAI Fall Symposium on Narrative Intelligence, 1999.*

[18] Meehan, J.R. (1977). Tale-spin, an interactive program that writes stories. *Proceedings of the 5th International Joint Conferences on Artificial Intelligence*, 1977. pp. 91-98.

[19] Mott, B., Callaway, C., Zettlemoyer, L., Lee, S., & Lester, J. (1999). Towards narrative-centered learning environments. In M. Mateas & P. Sengers (eds.) *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, 1999.

[20] Prince, G. (1987). *A Dictionary of Narratology*. Lincoln, NE: University of Nebraska Press.

[21] Sengers, P. (2000). Narrative intelligence. In K. Dautenhahn (ed.) *Human Cognition and Social Agent Technology*. John Benjamins Publishing.

[22] Sgouros, N.M. (1999). Dynamic generation, management and resolution of interactive plots. Artificial Intelligence, 107. 29-62.

[23] Thomas, F. & Johnson O. (1995). *The Illusion of Life: Disney Animation*. Hyperion.

[24] Weld, D.S. (1994). An introduction to least commitment planning. *AI Magazine*, Summer 1994.

[25] Young, R.M., Pollack, M.E., & Moore, J.D. (1994). Decomposition and causality in partial-order planning. In *Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems*, Chicago, IL, 1994. pp. 188-193.

[26] Young, R.M. (1994). A developer's guide to the Longbow discourse planning system. *Technical Report 94-4*, University of Pittsburg Intelligent Systems Program.

[27] Young, R.M. (1999). Notes on the Use of Plan Structures in the Creation of Interactive Plot. In M. Mateas & P. Sengers (eds.) *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, 1999.