Story and Discourse: A Bipartite Model of Narrative Generation in Virtual Worlds

R. MICHAEL YOUNG

Department of Computer Science
North Carolina State University
Raleigh, NC, 27695

young@csc.ncsu.edu
919.513.3038

In this paper, we set out a basic approach to the modeling of narrative in interactive virtual worlds. This approach adopts a bipartite model taken from narrative theory, in which narrative is composed of *story* and *discourse*. In our approach, story elements – plot and character – are defined in terms of plans that drive the dynamics of a virtual environment. Discourse elements – the narrative's communicative actions – are defined in terms of discourse plans whose communicative goals include conveying the story world plan's structure. To ground the model in computational terms, we provide examples from research under way in the Liquid Narrative Group involving the design of the Mimesis system, an architecture for intelligent interactive narrative incoroprating concepts from artificial intelligence, narrative theory, cognitive psychology and computational linguistics.

Interactive narrative, artificial intelligence, planning, cognitive models, suspense, discourse generation

Introduction

The number and type of computer system using interactive 3D interfaces continue to grow as the processing power of commercial graphics cards increases. While a significant portion of the most popular virtual worlds applications are in the $9 Billion per year interactive entertainment market, it is now common for users to interact with virtual worlds in applications ranging across simulation, training, education and social interaction. Many of these environments, especially those that are focused on entertainment, exploit informal adaptations of narrative techniques drawn from conventional narrative media in their design. Much of that work, however, conflates two central aspects of narrative structure that limit a) the range of techniques that can be brought to bear on the narrative's geneartion and b) the range of narrative structures that can be generated for a given environment. These two aspects of narrative are the structure of story and the structure of narrative discourse.

In this paper, we describe an approach to the generation of narrative-oriented interaction within virtual worlds that treats story and discourse as its two foundational levels. In this approach, we adapt models of narrative from narrative theory, computational linguistics and cognitive psychology, integrating these approaches with techniques from artificial intelligence in order to create interactive intelligent narrative virtual worlds systems.

The following section describes related work from both computer science and from narrative theory. Next, I give a brief introduction to Mimesis, the system used to create interactive virtual world applications, and describe the processes Mimesis uses for generating story-world and discourse-level narrative structure. Then I describe several example Mimesis plan fragments and show the means used to generate effective narrative structure from them. Finally, I characterize the role of story and discourse plans in several virtual world applications built using Mimesis, give a short description of the benefits of this approach to managing interactivity with our system, and describe work in progress that builds on story and discourse structure.

Related Work

Story and Discourse in Narrative Theory

The work described here adapts and extends existing work in artificial intelligence to account for specific story-oriented applications within 3D virtual environments. This approach is based on analytical methods first developed in narrative theory. Narratologists have provided an extensive characterization of narrative and its elements, describing the fundamental building blocks used by an author to create a compelling story [1,2,3]. Narrative-theoretic approaches, however, are analytic in nature and do not directly lend themselves to a computational model capable of being used in a generative capacity. A central challenge of any computational approach that seeks to operationalize concepts from narrative theory is to determine appropriate methods to translate concepts derived from analysis into concrete, formal models capable of being put to use in the creation of an interactive virtual environment.

While a broad range of approaches to the analysis of narrative exists, our work makes use of a structure that divides a narrative into two fundamental parts -- the *story* and the *discourse* [2,4] – and we construct distinct representations and tools to manage each. From a narratological perspective, a *story* consists of a complete conceptualization of the world in which the narrative is set. This includes all the characters, locations, conditions and actions or events that take place during the story's temporal extent. Two fundamental components of a narrative – its plot and its characters – are defined within the story itself. Distinct from the story, but closely tied to it, is the narrative *discourse*. Our discourse model represents those elements responsible for the *telling* of the story, rather than containing the story elements themselves. This notion of discourse differs from its conventional meaning. Specifically, the discourse we are generating is not communication between the user and the characters within the story. Rather, it is concerned with communication between the system and the user that conveys the storyline (which may include character dialog as individual elements).

In our approach, the construction of a narrative discourse can itself be divided into two conceptual aspects. One aspect is the determination of both the content of the discourse and its organization. To compose a narrative discourse, an author makes choices about those elements from the story to include in the story's telling and those elements to leave out. Further, the author determines additional information about the story-world to convey to the reader (e.g., properties of relevant objects, internal properties of the story's characters). Finally, the author must organize the discourse, determining what is to be told first, what second, etc, and how the sub-parts of the discourse should arranged so as to achieve the intended communicative effects on a reader.

A second aspect to the generation of narrative discourse is the selection of the specific communicative resources to be used to convey the story's elements to the reader. In a 3D virtual environment, these resources include a range of media, from voice-over narration to 3D camera control to background music. The work that we describe here focuses on the generation of coherent, cinematic camera control, though our results are applicable to aspects of communicative actions across media.

Computational Approaches to the Generation of Narrative

There are many examples of narrative-oriented interactive computer games, but the majority of this work involves interaction with storylines that are carefully crafted by game designers at design-time rather than genereted automatically at run-time. In contrast, several AI researchers have approaches the problem of narrative generation using techniques related to the approach we define here. Szilas [5], for instance, uses a bipartite system composed of a) a narrative logic component used to generate story elements and b) and a virtual narrator used essentially to filter the story elements to determine which actions are to be communicated to the user. Beyond this filtering process, however, no reasoning is performed to determine appropriate discourse structure.

Similarly, Cavazza and his collaborators [6] focus primarily on story structure rather than the discourse structure, using a hierarchical plan-based model similar to the one we describe below to control the characters within their narrative systems. In their recent work [7], they use a novel heauristic search algorithm for creating plans; the plans that they create, however, are roughly equivalent in representational

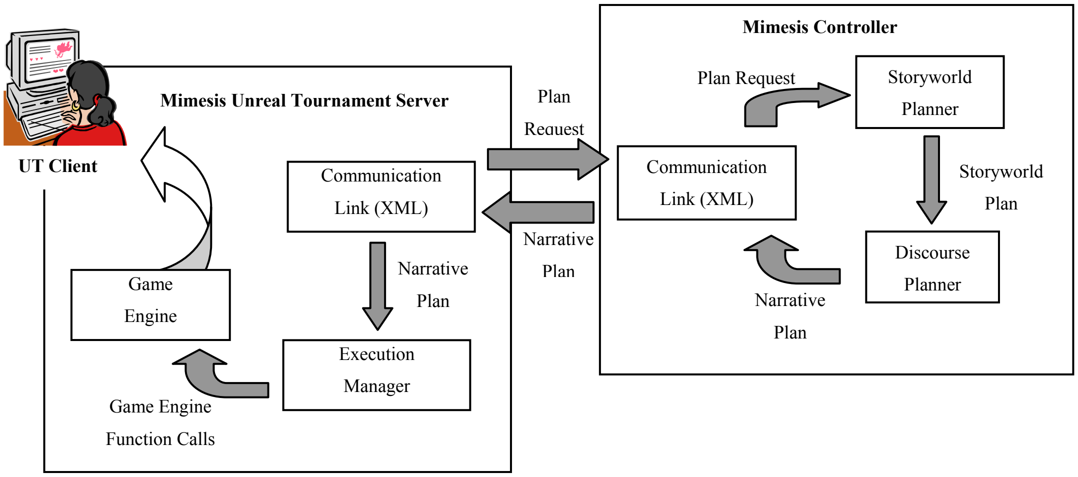

Figure 1. The Mimesis system architecture.

expressiveness to those used in their previous work.

Unlike plan-based approaches, Sgouros' work [8] uses an iterative approach to generate story actions. In his system, the action that occurs at a given moment is selected by a three step process. First, aspects of the story world context are considered by a rule set which generates a set of potential actions that could occur next. Second, these actions are filtered based on contextual weighting factors, and a single action is selected. Finally, the action is carried out, or resolved, within the system. This approach has two limitations. First, the creation of story elements is essentially opportunistic; actions are selected one at a time at the moment of their execution. Because there is no means for considering the relationships between current action choices and the execution of potential future actions, the degree of coherence of story actions may be quite limited.

Second, like much of the work related to narrative generation, Sgouros' work focuses on the generation of story-world actions to the exclusion of a sophisticated model of narrative discourse. Explicit description of the unfolding storyline is handled through pre-scripted text presented via pop-up dialog boxes or pre-scripted audio clips and character animations viewed via a user-controlled camera.

Generating Story and Discourse

Action and change are central to the nature of narrative. In most narratives, story-world action is initiated by the narrative's characters as they attempt to achieve their individual and collective goals. At the discourse level, a cinematographer acts in a goal-directed manner, intentionally composing shots and shot sequences to effectively communicate unfolding story action. This goal-oriented focus motivates us to use a *plan-based* model of the control of activity within virtual worlds; we have constructed an architecture, called Mimesis, that uses this model to generate plans for controlling characters operating within a narrative as well as for controlling media resources used for telling the narrative. We briefly describe the Mimesis architecture here. More details can be found in [9,10].

The Mimesis system integrates a suite of intelligent control tools with Unreal Tournament (UT), a commercially available 3D graphical game engine. While UT is well-suited as an engine for building conventional 3D interactive game titles, the representation of the environments that it models does not match well with those typically used by AI researchers. Like most virtual world engines, UT's internal representation is *procedural* -- it does not utilize any formal or declarative model of the characters, setting or the actions of the stories that take place within it. Consequently, direct integration of intelligent software components is not straightforward.

To facilitate this integration, Mimesis overrides UT's default mechanisms for controlling its virtual environment, using instead a client/server architecture in which low-level control of the game environment is performed by a customized version of the game engine (called the *Mimesis Unreal Tournament Server,* or *MUTS*) and high-level reasoning about narrative structure and user interaction is performed remotely by an intelligent control element (called the *Mimesis Controller,* or *MC*). The architecture is presented in Figure 1 and described briefly here. In later sections, we characterize several example virtual world applications built for and controlled by the Mimesis architecture.

Within Mimesis, the MC acts essentially as a narrative server, determining the narrative elements of the user's experience within the virtual world. The MC is responsible for

- the generation of a story (in the form of a *story-world plan* characterizing all character actions that are to be performed within the environment)
- the generation of a *discourse plan* characterizing the media-specific communicative actions used to convey the story to the user, and
- the maintenance of a coherent narrative experience in the face of unanticipated user activity.

At start-up, the MUTS sends a message to the MC requesting a story. This request identifies a goal state for the story, the MUTS' current world state and the library of actions that are available for characters in the MUTS' world. The MC then generates a story-world plan [11] that describes all the actions that the characters will execute in the story world. It sends this plan as input to the discourse planner, which generates a specification of the communicative action (in our case, 3D camera shot specifications) that will convey the elements of the story plan to the user. These two plans are integrated, and the resulting plan is encoded into an XML message and transmitted to the MUTS via a socket connection.

Upon receiving the XML message, the *Execution Manager*, the element within MUTS responsible for driving the story's action, builds a directed acyclic graph whose nodes represent individual actions in the plans and whose arcs define temporal constraints between actions' orderings. The Execution Manager uses one-to-one mappings from the action types of the nodes in this graph to game engine functions and from the parameters of each action to instances of game engine objects in order to construct function calls that will drive the appropriate animations and state changes within the virtual world. The structures created and used by these elements are described in more detail in the following sections.

Creating the Story-World Plan

The plan structures that we employ are produced by the DPOCL (Decompositional Partial-Order Causal Link) planner [12]. DPOCL plans are composed of *steps* corresponding to the actions that characters carry out within a story; in DPOCL,

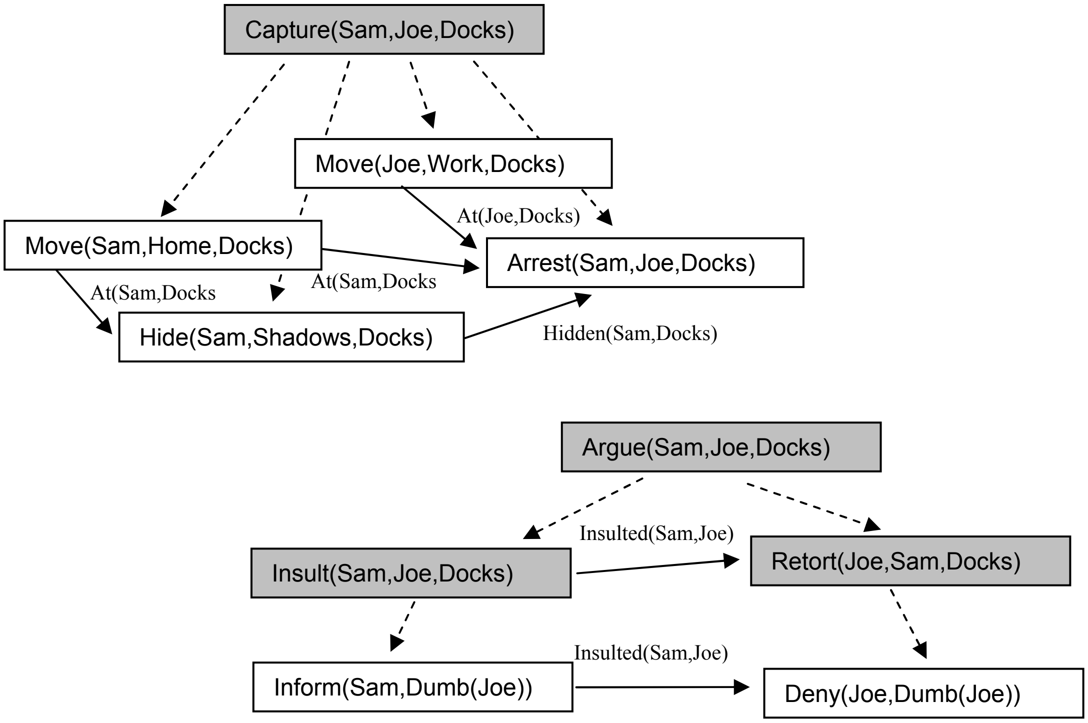

Figure 2. A DPOCL plan fragment, meant to be part of a larger story-plan structure. White boxes indicate primitive actions, gray boxes indicate abstract actions, dashed arcs indicate subplan relationships, solid arcs indicate causal links from effects of one action to the preconditions of another. Temporal ordering is indicated in rough left to right order.

each step is defined by a set of *preconditions*, the conditions in the world that must hold immediately prior to the step's execution in order for the step to succeed, and a set of *effects*, the conditions in the world that are altered by the successful execution of the action. In addition to a set of steps, a DPOCL plan contains a set of *temporal constraints* defining a partial temporal ordering on the execution of the plan's steps and a set of *causal links* connecting pairs of steps. Two steps s_1 and s_2 are connected by a causal link with associated condition c (written $s_1 \rightarrow^{c} s_2$) just when c is an effect of s_1 and a precondition of s_2 and the establishment of c by s_1 is used in the plan to ensure that c holds at s_2.

Further, DPOCL plans contain information about the hierarchical structure of a plan, similar to the representation used by hierarchical task network (HTN) planners [13].

Because action sequences within narratives are often *episodic* that is, because they follow common patterns of action, these hierarchical structures are particularly amenable to representing story fragments. A DPOCL plan fragment is shown in Figure 2.

Adopting a plan-based model of story structure allows the system to compose new stories in response to novel starting states or goal specifications, or to customize a story based on a user's interests and knowledge. The use of DPOCL plans has two additional advantages. First, the formal properties of the planning algorithm guarantee that the plans contain adequate structure to effectively control the story world's virtual environment. Specifically, DPOCL plans are provably *sound*, that is, when executed, each action in them is guaranteed to execute correctly and the plans themselves are guaranteed to achieve their top-level goals. These properties of DPOCL make the plans it produces well-suited for use in controlling the execution of a virtual environment [6,11].

A second benefit to the use of plans to drive a narrative is in the plan's structural correspondence to a user's mental model of the story it defines. Recent research [14,15] suggests that hierarchical causal link plans like DPOCLs, as well as the techniques used by the DPOCL algorithm to create them, make for effective models of human plan reasoning. Our empirical studies indicates that the core elements of DPOCL plans match up with the models of narrative structure defined and validated by psychologists [16]. By using a formal representation for story structure that corresponds to users' models of stories, we can make more direct predictions about the users' understanding of the stories we create. We rely on this correspondence when designing techniques to create specific narrative effects, such as the models of suspense discussed in the following sections.

Creating the Narrative's Discourse

A narrative system must not only create engaging story-world plans, it must use its resources to tell the story effectively. In this paper we discuss one particular strategy used in the effective creation of a narrative: building narrative discourse involves the central task of determining the content and organization of a sequence of camera shots that film the action unfolding within a story world.

In the work described here, we build on our previous research on the generation of natural language discourse to generate discourse plans for controlling an automated camera that is filming the unfolding action within a 3D story-world. To create these discourse plans, we use a discourse planning system named Longbow [17]. The Longbow planner is built on the core DPOCL algorithm, and so the two planners' representations are quite similar. In our approach, 3D camera shots and shot sequences are viewed as planned, intentional action whose effects obtain in the cognitive state of the user. Individual camera shots are treated as primitive communicative actions, multi-shot sequences and cinematographic idioms are characterized using hierarchical plan operators, and, as in conventional discourse planning, plan structure that specifies the communicative content of a discourse is created to achieve particular effects upon the mental state of the user.

Conventional discourse planners take as input a set of propositions intended to be conveyed to the user of a system, along with a model of the user's existing knowledge of the domain of discourse and a library of plan operators describing both the primitive communication actions available to the planner (e.g., typically speech acts such as INFORM or REQUEST) and definitions for a set of abstract actions and their sub-plan schemas, sometimes referred to as *recipes*. Abstract operators often specify rhetorical structure [18,19] in a discourse (e.g., when one part of a discourse stands as evidence for the claim set forth in a second part of a discourse) and their sub-plan schemas specify how more primitive collections of communicative actions can be combined to achieve the abstract act's communicative effects.

There are several important ways that the task of narrative discourse generation – and our approach to it – differ from the task of discourse generation in conventional contexts. In our approach, the propositional content that the narrative discourse planner receives as input refers not just to relations that hold in the domain of discourse, but also to propositions describing the structure of the story-world plan. For instance, in addition to generating discourse that conveys the fact that a character has a gun, the narrative discourse must also convey the action of the character using the gun to rob a bank. The task of the discourse planner is, in part, to generate camera action sequences that convey the execution of story-world plan actions to the user.

Beyond the requirement to communicate a different type of content in narrative discourse, our approach to the generation of plans for 3D narrative discourse addresses two key problems. First, the narrative discourse that we generate must contain structure beyond that which simply mirrors the structure of the actions executed in the story world. Cinematic discourse contains both rhetorical structure, aimed at conveying propositions about the story world to a user, but also idiomatic structure mirroring the use of patterns for shot composition used in film [20,21]. Our plan operators these aspects of discourse structure and combine them effectively to tell the story. Consider the schematic of a hierarchical discourse plan operator shown here:

```
establish_scene_change(current_action)
constraints:
        previous_action(current_action, previous_action)
        location(previous_action,previous_location)
        current_action_being_filmed(current_action)
        location(current_action,current_location)
        not(equal(current_location,previous_location))
        participants(current_action, agent1, agent2)
        location(agent1,current_location)
        location(agent2,current_location)
        surrounding_location(exterior,current_location)
        not(equal(exterior, previous_location))
substeps:
        establish_exterior_location(exterior)
        establish_character_location(agent1,agent1)
ordering:
        establish_exterior_location < establish_character_location
```

This operator provides an abstract description of the communicative actions involved in conveying a scene change that also involves a change of location. The constraints of the operator, represented as a set of logical clauses, bind variables (indicated in italics) to entities within the story-world and its plan. The operator is applicable only when all its constraints can be satisfied. These constraints serve to a) pick out the previously filmed action, and the location for both this past action and the next action to be filmed, b) ensure that the locations of the two actions are distinct, c) pick out the two characters involved in the new action to be filmed and make sure that they are both located in the same place, d) and finally pick out a larger location that surrounds the characters' location, checking to make sure that this surrounding location is not the same as the location of the previous action.

The sub-steps created in the discourse plan involve filming the surrounding location and filming the location of the two characters. A temporal ordering imposed by the operator requires the shot filming the surrounding location to occur before the shot that shows the location of the characters. These two sub-steps are themselves abstract, and a plan that contains them will make context-specific choices about the ways in which they can be further refined, eventually creating specific constraints for camera actions. For instance, an `establish_character_location` action might be refined into one long shot in which both characters and their setting can be seen. Alternatively, it might be refined into a sequence of two shots, one medium shot where the first character can be seen, with her location visible in the background and a shot of the second character, with the first character visible in the background. By placing the appropriate constraints on the operators that create the hierarchical structure of the discourse plan, we can effectively embed the rhetorical structure of a film into the plan for filming the story-world's actions.

A second key problem addressed by our approach to discourse planning is the temporal integration of the story-world and discourse-level plans. The actions in discourse plans for narrative in virtual worlds, unlike actions in plans for textual narrative, must themselves execute. Camera actions for panning, tracking, fading, etc, all require time to play out, a physical location from which the camera films, physical objects that must be included or excluded from the field of view, etc. A particularly complicating aspect of this is that these camera actions must execute in the same temporal and spatial environment as the objects of the story that they must convey to the user. A knowledge representation for narrative discourse must take this shared evironment into account or risk creating suboptimal (or even inconsistent) plans.

For instance, consider the case where camera action C_1 is responsible for filming action s_1 and camera action C_2 is responsible for filming action s_2. If s_1 completes its execution prior to s_2 beginning, then C_1 must complete its execution prior to C_2. In plans where the successful execution of C_1 depends upon the user knowing some property of the domain first established in C_2, the inconsistency must be detected and remedied. If s_1 and s_2 happen concurrently in the same spatial location, then C_1 and C_2 could be replaced by a single appropriately placed shot C_3. Without considering their colocation/co-occurrence, a planner would not be able to generate this option.

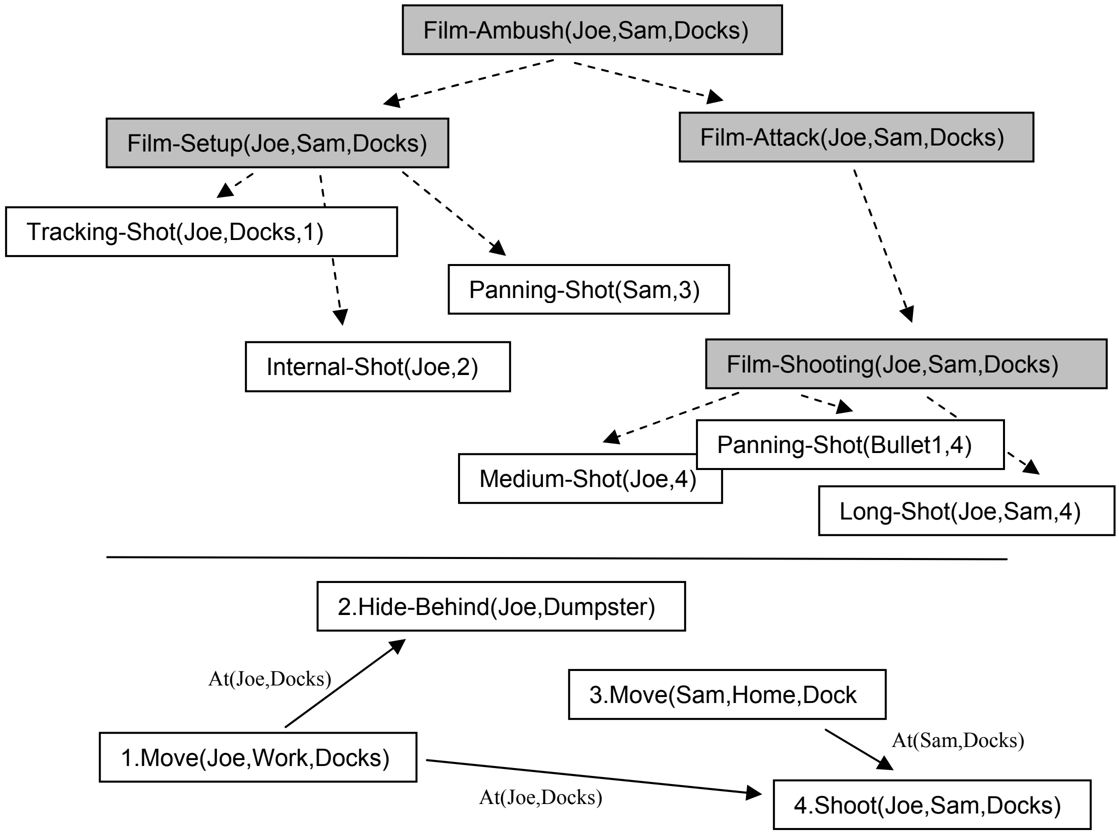

Figure 3. A narrative plan fragment showing both DPOCL and Longbow plan structures. The DPOCL plan fragment, shown below the dividing line, represents a simple plan for an ambush between two characters, Joe and Sam. The Longbow plan fragment is shown above the horizontal divider, and represents the plan for controlling the camera. For clarity, DPOCL plan steps have been given unique numeric labels. Arguments to camera actions specify both the storyworld objects to be filmed as well as the storyworld action during which the filming must occur (indicated by corresponding numeric labels).

In order to allow the operator writer to specify the temporal relationships between the execution of camera actions and the story-world actions that they must film, primitive camera actions in the discourse planner can be annotated with temporal constraints between the two plans.

Constraint	Explanation
`begin_before_start(swa)`	Begin current action any time before start of storyworld action *swa*.
`begin_at_start(swa)`	Begin current action at the same time as the start of storyworld action *swa*.
`begin_after_start(swa)`	Begin current action any time after the start of storyworld action *swa*.
`begin_before_end(swa)`	Begin current action any time before the end of storyworld action *swa*.
`begin_at_end(swa)`	Begin current action at the same time that storyworld action *swa* ends.
`begin_after_end(swa)`	Begin current action any time after the end of storyworld action *swa*.
`end_before_start(swa)`	End current action any time before the start of storyworld action *swa*.
`end_at_start(swa)`	End current action at the same time as the start of storyworld action *swa*.
`end_after_start(swa)`	End current action any time after the start of storyworld action *swa*.
`end_before_end(swa)`	End current action any time before the end of storyworld action *swa*.
`end_at_end(swa)`	End current action at the same time that storyworld action *swa* ends.
`end_after_end(swa)`	End current action anytime after storyworld action *swa* ends.

Table 1. Temporal constraints relating primitive camera actions to the execution of the storyworld actions that they film. In these actions, *swa* is a variable bound to the story-world action being filmed.

These constraints relate the start and end times of the camera actions to the start and end times of the actions that they film. These constraints are listed in Table 1. Constraints can be composed using logical operators (e.g., and, not, or) to create complex temporal relationships between filming and acting.

Suspense and Narrative Plans

By relating a precise computational model of action to a mental model of narrative, we are able to make predictions about the cognitive and affective consequences to a user that is experiencing the execution of specific kinds of plan structures. For instance, we have been exploring the role of plans and planning on suspense, an essential property of conventional narrative forms such as the film or novel. While suspense can be of many forms and arise in many kinds of situations, we have focused on suspense deriving from a user's knowledge of the unfolding plans of a narrative's characters. In particular, we focus on suspense arising from the anticipation of future events and their consequences on the goals of the narrative's protagonist (whether that protagonist is the user or another character within the narrative).

Recent work in cognitive psychology [22,23,24,25] has considered the role of narrative structure in the creation and maintenance of this type of suspense in film and literature. Gerrig and Bernardo [23] suggest that people who read fiction act as problem-solvers, continuously looking for solutions to the plot-based dilemmas faced by the characters in a story-world. Their work indicates that a reader's suspense is dependent upon the number of solutions that she can find to the protagonist's problem: suspense is greater when there are fewer solutions accessible.

Our approach approximates the problem solving activity that a user performs when seeking solutions to plot-related problems as planning in a space of story-world plans. Our cognitive model employs the model of planning as *refinement search* defined by Khambamphati, *et al* [26]. A refinement planning algorithm represents the space of plans that it searches using a directed graph; each node in the graph is a (possibly partial) plan. An arc from one node to the next indicates that the second node is a refinement of the first (that is, the plan associated with the second node is constructed by repairing some flaw present in the plan associated with the first node). In typical refinement search algorithms, the root node of the plan space graph is the empty plan containing just the initial state description and the list of goals that together specify the planning problem. Nodes in the interior of the graph correspond to partial plans and terminal nodes in the graph are identified with completed plans (solutions to the planning problem) or plans that cannot be further refined due for instance, to inconsistencies within the plans that the algorithm cannot resolve.

We have successfully used this model to approximate the plan-based reasoning performed by readers when understanding instructional texts [15]. By characterizing the space of plans that a user might consider when solving problems faced by a protagonist at a given point in a plot, the model can be used to make predictions about the amount of suspense a user will experience at that point. To do this, we model the set of beliefs held by a user as she experiences an unfolding narrative. At any point in the narrative, a user will have a set of beliefs about that state of the story world, a set of beliefs about the goals of the story's protagonist and a set of beliefs about the action operators available to the characters acting within the story. These three elements are used to create a DPOCL *planning problem*, the specification used as input to DPOCL to create a plan (in this case, a plan that solves the protagonist's goals given the story world's current state as known by the user). DPOCL's refinement search algorithm creates not just a single plan that solves the protagonist's goals, but the space of possible plans given the problem specification. To the extent that DPOCL's configuration mirrors the user's planning process, this plan space approximates the space of solutions considered by the user when searching for solutions to the protagonist's current problems.

To determine the level of suspense a user may experience at a particular point in a story, we have developed and empirically evaluated a model [16] that relates characteristics of this space to the psychological results described above. For instance, when there are a large number of successful solution plans at the leaves of the plan space graph, our model correctly predicts that a user's experience of suspense would be lower than when there were few successful solutions to the current planning problem. Additional features of this space, for instance, the ratio of failed plans to total plans, also prove to be an element of an effective prediction of suspense. As a result of initial experimental results, we are considering extending the representation of plans to include additional features, such as the perceived probabilities of success for each of the plans, and are evaluating those new features for their role in the problem-solving process of users.

Results from this work also suggests means by which suspense can be increased or decreased. The features of a plan space that a user considers will differ depending on the planning problem she is attempting to solve. By controlling what facts the user believe s about the story world at a given time, for instance, by generating camera sequences that explicitly convey some facts while explicitly eliding others, the system can, in effect, define a planning problem with the suspense properties appropriate for each point in the narrative.

The Challenges of Interactivity in Story and Discourse

Control and Coherence

Our work described above takes an idealized stance in which the user is not accounted for except as a passive observer. While this assumption is entirely valid for conventional narrative media such as film or literature, the assumption is almost always invalid for narrative-oriented virtual worlds. A key feature of these worlds is the level of interactivity that they offer the user. The ability to step into the narrative world and play a character in the story, to take substantive action within the unfolding story, is a key distinguishing feature of virtual worlds and stories.

A central issue in the development of effective and engaging interactive narrative environments is the balance between coherence and control. The understandability of any narrative is determined, in part, by it's *coherence*, that is, by the user's ability to comprehend the relationships between the events in the story, both within the story world (e.g., the causal or temporal relations between actions) and in the story's telling (e.g., the selection of camera sequences used to convey the action to the user). Dramatists often refer to narrative as having a premise or point; stories are told for a reason and much of our comprehension of a story involves the construction of cognitive models that predict or explain these relationships. Systems that construct actions for telling a story should respect the story’s coherence by clearly linking each action in the narrative to its overall structure.

The degree of engagement by a user within an interactive narrative lies, to a great extent, with the user’s perceived degree of *control* over her character as she operates within the environment. The greater the user’s sense of control over her character, the greater will be her sense of presence [27], that is, the sense that she is a part of the story world and free to pursue her own goals and desires. Unfortunately, control and coherence are often in direct conflict in an interactive narrative system. To present a coherent narrative, the actions within an interactive narrative are carefully structured (either at design time by human designers, in the case of conventional computer games, or at run time by intelligent systems like the one described here) so that actions at one point in the story lead clearly to state changes necessitated by actions occurring at subsequent points in the story. When users exercise a high degree of control within the environment, it is likely that their actions will change the state of the world in ways that may interfere with the causal dependencies between actions as intended within a storyline.

Conventional forms of narrative (e.g. film and novel) resolve the issue of coherence versus control by completely eliminating control; the audience is a passive observer. Computer game developers, in contrast to film makers, introduce interactivity in their systems, but carefully limit the control exercised by the user by designing the environment so that the user’s choices for action at any point reduce to a small set of options moving the user through a pre-defined branching structure. In the remainder of this paper, we discuss a technique used in the Mimesis architecture called narrative mediation which allows a degree of control and coherence that lies between that of computer games and conventional narrative media.

Managing Interactivity

As described above, Mimesis drives the action within its story world based on the structure of a plan produced by a narrative planner. As users issue commands for their characters to perform actions within the story world, these actions have the potential to undo conditions in the world that are critical to the success of actions in the narrative plan that have not yet executed. Consequently, before carrying out directives from the user, the corresponding actions must be checked against the narrative plan to determine how they fit with the plan's structure. This is accomplished by relating each input command from the user's keyboard or mouse activity to some predefined action α specified by a plan operator.

Each action α performed by the user is automatically characterized in one of three ways with respect to the unexecuted portion of the plan. One possibility is that α is *constituent* to the plan – α matches an action prescribed by the narrative plan for execution by the user, in which case the user is doing exactly the action that the system desires her to do in order to perform that portion of the storyline. The second possibility is that α is *consistent* with the plan – α is not constituent and none of the effects of α interact with any of the plan's remaining structure. For example, it may be consistent if the user rotates her character in a circle in order to orient herself spatially before walking out of a room, as long as her act of walking out of the room is part of the narrative and is successfully performed during the appropriate timeframe. The third possibility is that α is *exceptional* – α is not constituent and one or more of α's effects threaten the conditions in the world required by future agent actions. Specifically, an exception occurs whenever a user attempts to perform some action α, where some effect ¬*e* of α threatens to undo some causal link *s*1 e *s*2 between two steps, *s*1 and *s*2, with condition *e*, where *s*1 has occurred prior to α and *s*2 has yet to occur.

If a user performs an exceptional action, the effects of the exception on the virtual world undoes the condition of at least one causal link in the plan, invalidating some or all of the plan's subsequent structure. It is the responsibility of the system to detect exceptions when they arise and to respond accordingly in a manner that balances the need to preserve the coherence of the narrative with the need to preserve the user's sense of control. Within the Mimesis system, response to exceptions occurs in one two ways. Either the system allows the exception to occur and restructures the narrative plan mid-story, or it prevents the exception from actually executing, in effect coercing the user into compliance with the existing plan structure. We refer to this process of exception detection and response as *narrative mediation*.

The most straightforward response to an exception is via *intervention*. Typically, the success or failure of an action within a virtual environment is determined by function calls that approximate the physical rules of the underlying story world (e.g., setting a nuclear reactor's control dial to a particular setting may cause the reactor to overload). However, when a user's action would violate one of the story plan's constraints, Mimesis can intervene, causing the action to fail to execute. In the reactor example, this might be achieved by surreptitiously substituting an alternate set of action effects for execution, one in which the "natural" outcome is consistent with the existing plan's constraints. A control dial momentarily jamming, for instance, will preserve the apparent consistency of the user's interaction while also maintaining safe energy levels in the story world's reactor system.

The second response to an exception is to adjust the narrative structure of the plan to accommodate the new activity of the user. The resolution of the conflict caused by the exception may involve only minor restructuring of the narrative, for instance, selecting a different but compatible location for an event when the user takes an unexpected turn down a new path. Accommodation may involve more substantive changes to the story plan, however, and these types of modifications can be computationally expensive. For instance, should a user instigate a fight with a character that is intended to be a key ally later in the story or unintentionally destroy a device required to rescue a narrative's central character, considerable re-planning will be required on the part of the MC's narrative planner.

In order to handle exceptions in an interactive narrative system, the narrative planner analyzes its plans prior to execution. Analysis begins at the start of the plan and proceeds forward in time in discrete steps corresponding to the execution of each action in the plan. Because the plan structure contains explicit representations of all causal and temporal dependencies between the plan's steps, it is possible to examine each world state between actions, looking for points where enabled user actions (that is, actions whose preconditions for execution are satisfied at that point in the plan) can threaten the plan structure. When potential exceptions are identified, the planner weighs the computational cost of re-planning required by accommodation against the potential cost incurred when intervention breaks the user's sense of agency in the virtual world. The reader is encouraged to see [10] for a discussion of the approach used for narrative mediation within Mimesis, including techniques for pre-computing responses to exceptions in order to increase system response time.

Narrative mediation has already proven useful with story-world actions; the efficacy of the technique comes in part because of the nature of user actions that might raise exceptions. Exceptions can typically be identified with discrete and instantanous user activity (e.g., firing a laser blaster, starting a car engine) rather than with continuous with substantial duration (e.g., capturing a space station, driving to Pittsburgh). Further, most story-world actions that a user will perform are going to be consistent rather than exceptional. Because of these two features, exceptions are straightforward to identify and rarely place high computational demands on the system. In contrast, detecting and responding to exceptions at the level of narrarive disocurse is more difficult. Camera actions are inherently continuous, making it difficult to determine what the current effects of a camera movement will be. Futher, camera actions happen almost all the time inside most game-related virtual worlds.

To date, we have avoided this issue by using mediation techniques to contexts where a user's camera control is explicitly limited (e.g., in cut scenes where the user has a menu for controlling camera views). To address, the problem, however, we are building a component that monitors features of a user's field of view (e.g., what is the level of lighting, what entities are in or out of view) and the motion of the camera to anticipate changes in camera position that may violate the structure of the discourse plan. This extension will take into account the changes in knowledge state caused by a user's choice of shot composition; by integrating this model into a discourse-level mediation scheme, we expect that techniques similar to the accomodation approach described in the previous section will be applicable. However, a discourse-level equivalent to intervention seems more problematic. Because camera control is tightly coupled with a user's input actions (e.g., mouse manipulation), an approach that substitutes alternate camera control actions without disrupting a user's sense of agency in the virtual world is a challenge to define.

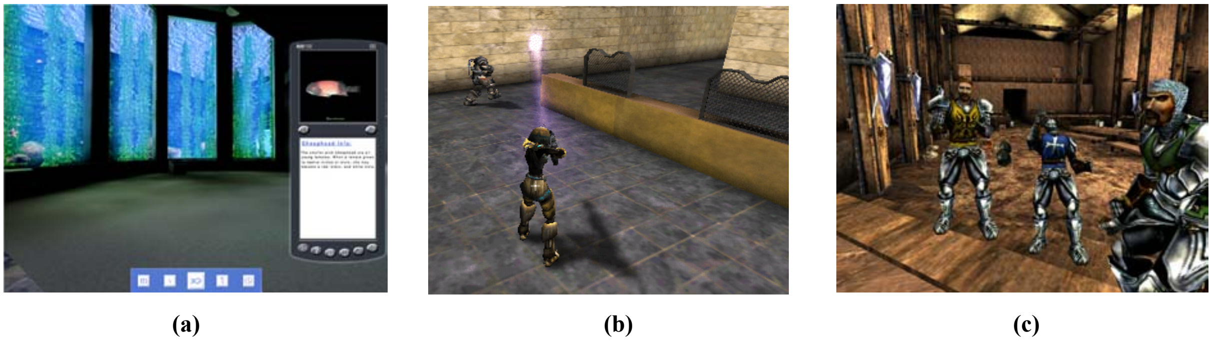
(a) (b) (c)

Figure 4. The images above show example worlds controlled by Mimesis, including (a) a virtual tour of the Monterey Bay Aquarium, (b) a futuristic bank robbery story and (c) the great hall of a medieval castle.

System Performance Summary

To date, we have constructed several test-bed environments for use with Mimesis; performance results from these experiments have been encouraging. The number of characters in these worlds have ranged from two to 20, including a single user. Because characters are treated by the planners like any other resouce available within the story world, the number of characters or users in the story is not a limiting factor for the planner's performance. Rather, the number and complexity of the actions in the operator library are more often the determining factor in the speed of the planners' execution.

We currently generate plans for several test problem domains, described briefly below. These planning problems include forty to sixty action operators in their operator libraries and goal expressions containing five to fifteen predicates. The planning process considers roughly three thousand alternate plans during plan generation, running in under a half second on a Pentium 4 1GHz. Typical plans that are produced contain about 60 steps. While this length is not sufficient to capture an entire story, future tests will expand the complexity of the planning task to generate longer action sequences.

Performance for DPOCL and Longbow planning is roughly identical given comparably complex planning problems. Integration between discourse and story plans serves to improve overall performance, since the addition of temporal constraints between the two plan structures can serve to filter out plans that are temporally inconsistent earlier in the planning process, cutting down on the extra planning work that would have been performed on the unexecutable plans.

Performance of the narrative mediation component has also been encouraging, although pre-computing and caching techniques are required in order for this technique to be feasible. In our approach, a recursive algorithm is run on the initially created narrative plan in order to determine all potential responses to exceptions that might arise. For those exceptions that require accomodation, replanning is done at that point in the plan, determining appropriate new plans to put in place should the exception arise. These new plans are then analyzed for exceptions, and so on until a time-limit is reached. This approach requires a higher start-up time, but improves overall system response time as exceptions arise. Results of these computations are cached in the MUTS, facilitating quick lookup. When the MC's processor load is low during execution, for instance, when the user is idle, additional pre-computing can be done, anticipating responses to exceptions at a greater look-ahead depth.

Summary and Conclusions

In this paper, we have set out a basic approach to the modeling of narrative in interactive virtual worlds. This approach adopts a bipartite model of narrative as story and discourse in which story elements – plot and character – are defined in terms of plans that drive the dynamics of a virtual environment. Narrative discourse elements – the narrative's communicative actions – are defined in terms of discourse plans whose communicative goals include conveying the story world plan's structure. To ground the model in computational terms, we have provided examples from the Mimesis system, including features of the system that have already been implemented and features that are under development, targetted at the theoretical division we have set out above. While there are many possible means to approach a story-and-discourse model of interactive narrative, our goal is to demonstrate the effectiveness of this model using the Mimesis system as a testbed; our initial results, mentioned here and in the work we cite, are encouraging.

Acknowledgements

We are grateful to the many students that have contributed to the Liquid Narrative Group and the development of the Mimesis system. Significant contributors to the work described in this paper include Dan Amerson, Mark Branly, D'Arcey Carol, Arnav Jhala, Shaun Kime, Milind Kulkarni, R.J. Martin, Mark Riedl, C.J. Saretto and Sirish Somanchi. This work was supported by National Science Foundation CAREER award #0092586.

References

[1] Bal, M. Narratology: An Introduction to the Theory of Narrative, University of Toronto Press 1998.

[2] Chatman, S. Story and Discourse: Narrative Structure in Fiction and Film, Cornell University Press 1990.

[3] Rimmon-Keenan, S. Narrative Fiction: Contemporary Poetics, Routledge, 2002.

[4] Emmot, K. Narrative comprehension: A Discourse Perspective, Oxford, 1999.

[5] Szilas, N. A New Approach to Interactive Drama: From Intelligent Characters to an Intelligent Virtual Narrator, in the Working Notes of the AAAI Spring Syposium on AI and Interactive Entertainment, Stanford, CA, 2001.

[6] Cavazza, M., Charles, F., Mead, S., Planning characters' behaviour in interactive storytelling. Journal of Visualization and Computer Animation, 13(2): 121-131, 2002

[7] Fred Charles, F. , Lozano, M. Mead, S., Bisquerra, A., and Cavazza, M. , Planning Formalisms and Authoring in Interactive Storytelling, in the Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment, 2003.

[8] Sgouros, N. M., "Dynamic Generation, Management and Resolution of Interactive Plots," Artificial Intelligence, vol. 107, no. 1, pp: 29-62, 1999.

[9] Young, R. M. And Riedl, M. Towards an Architecture for Intelligent Control of Narrative in Interactive Virtual Worlds. In: Proceedings of the International Conference on Intelligent User Interfaces, January, 2003.

[10] Riedl, M., Saretto, C.J. and Young, R. M. 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, June, 2003.

[11] Young, R. M. Notes on the Use of Plan Structures in the Creation of Interactive Plot. In: *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, Cape Cod, MA, 1999.

[12] Young, R. Michael, Martha E. Pollack and Johanna D. Moore, Decomposition and causality in partial-order planning, in Proceedings of the Second International Conference on AI and Planning Systems, Chicago, IL, pages 188-193, July, 1994.

[13] Sacerdoti, E. A Structure for plans and behavior. Elsevier, New York, 1977.

[14] Rattermann, M. J., Spector, L., Grafman, J., Levin, H. and Harward, H. Partial and total-order planning: evidence from normal and prefrontally damaged populations, Cognitive Science 2002; 25(6); 941-975.

[15] Young, R.M. Cooperative Plan Identification: Constructing Concise and Effective Plan Descriptions. Proceedings of the National Conference of the American Association for Artificial Intelligence, pages 597-604. Orlando, FL, 1999.

[16] Christian, D. and Young**,** R. M. Comparing Cognitive and Computational Models of Narrative Structure**.** Liquid Narrative Technical Report TR03-001, Liquid Narrative Group, Department of Computer Science, North Carolina State University, Raleigh, NC. 2003.

[17] Young, R. M., Moore, J. D., and Pollack, M., Towards a Principled Representation of Discourse Plans, In the proceedings of the Sixteenth Conference of the Cognitive Science Society , Atlanta, GA, 1994.

[18] Mann and Thompson. Rhetorical structure theory: A theory of text organization. In Livia Polyani, editor, The Structure of Discourse. Ablex Publishing Corporation, 1987.

[19] Moore, J. and Paris, C. Planning text for advisory dialogues: Capturing intentional and rhetorical structure, Computational Linguistics 1993; 19(4): 651-695.

[20] Arijon, Daniel. Grammar of the film language, Communication Arts Books, Hastings House: New York, 1976

[21] Monaco, J. How to Read a Film: The Art, Technology, Language, History and Theory of Film and Media, Oxford University Press, 1981.

[22] T. Trabasso and L. Sperry. Causal relatedness and importance of story events. Journal of Memory and Language, 1985, 24:595-611.

[23] Gerrig, R., and Bernardo, D. Readers as problem-solvers in the experience of suspense. Poetics 1994; 22; 459-472.

[24] Ohler, P. and Niedling, G. Cognitive modeling of suspense-inducing structures in narrative films. In Suspense: Conceptualizations, theoretical analyses and empirical explorations. Vorderer, et al, eds, Lawrence Erlbaum. 1996.

[25] Comisky, P., and Bryant, J. Factors involved in generating suspense. Human Communication Research 1982; 9(1):49-58.

[26] Khambamphati S, Knoblock, C and Yang, Q., Planning as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial-Order Planning in Artificial Intelligence vol. 76(1-2), 167-238, 1995.

[27] Lombard, M. & Ditton, T. At the heart of it all: the concept of presence. Journal of Computer-Mediated Communication 1997; 3(2).