Story Representation and Interactive Drama

Brian Magerko

University of Michigan
1101 Beal Ave., Ann Arbor, MI 48109
brian@magerko.org

Abstract

When building a story-intensive game, there is always the question of how much freedom to give the player. Give the player too little, and he may feel constrained and disconnected from the character he is controlling. Give him too much freedom, and the progression of the story may lag or stop altogether. The field of i*nteractive drama* attempts to strike a balance between interaction and authorship. The story experienced in an interactive drama is dependent both on the plot content authored as well as the player's choices in the story. Which story representation is appropriate for a particular approach to interactive drama and the relationship between that language and other elements of the architecture is a key factor in design. This paper introduces our interactive drama architecture, IDA, and addresses the requirements it has for a story representation. How those requirements are met by our representational choices is the focus of the rest of the paper.

Introduction

Traditional computer games rarely tell a deep and well-developed story. When a game does attempt to incorporate plot into the game experience, the player's dramatic experience is usually heavily constrained to match a fairly linear set of dramatic events; there is little, if any, customization of the narrative to fit an individual player's experience. In the field of *interactive drama*, we try to bridge a connection between player desires and story content to provide a deep connection between what the player does in the story world and where the story leads. Our view of what interactive drama is closely follows the definition given by Brenda Laurel:

> An "interactive drama," then, is a first-person experience within a fantasy world, in which the User may create, enact, and observe a character whose choices and actions affect the course of events just as they might in a play. The structure of the system proposed in the study utilizes a playwriting expert system that enables first-person participation of the User in the development of the story or plot, and orchestrates system- controlled events and characters so as to move the action forward in a dramatically interesting way (Laurel 1986).

Our approach to interactive drama is author-centric; we view this medium as the means for a human author to communicate an artistic vision. If the human player is going to contribute to the plot is some meaningful fashion, they can do so within the narrative boundaries, the *story space* (i.e. the space of possible story experiences), that is defined by the author. Thus, a generic interactive drama is comprised of the following features: the player, a story world for the story to take place, characters to perform the story, an author, a story representation for the author to use, and the storytelling mechanism. The player interacts with the synthetic characters in the story world, experiencing a narrative defined by the authored story space. The author writes story content in some logical representation, which is an input to the storytelling mechanism. The story representation should allow the author to be as specific or abstract as he desires in terms of plot content and the presentation of that content. The purpose of the storytelling mechanism is to help move the story along. This mechanism could be the autonomous behaviors of the synthetic characters, a story director that manages the actions of the characters in response to player actions, or even a set of rules that help generate dramatic content.

There have been several approaches to creating interactive drama systems, each with their own needs for story representation. For example, the Liquid Narrative group's system, MIMESIS, relies on a STRIPS-style planning representation to support replanning (Young el at. 2004). The University of Teesside's system, a heavily character-based approach, represents story as the set of HTN's that define the characters (Cavazza et al. 2002). The ALT-SIM project, developed at the Institute for Creative Technologies, uses logical formalizations of commonsense psychology to encode the entire player mental experience (Gordon and Iuppa 2003). Yet another system, Façade, represents story as annotated story beats, which are heuristically chosen during the course of the drama in response to the player's interactions with the system (Mateas and Stern 2002). Each of these systems, as well as others not mentioned here, represent steps away from the typical story graphs that have been traditionally used in interactive media.

An Interactive Drama Architecture

The system described in this paper, IDA (Interactive Drama Architecture) (Magerko et al. 2004; Magerko and Laird 2004), is a step towards providing and making use of such a representation. As shown in Figure 1, IDA is comprised of the player, the human author, the director and actor agents, and the virtual world that the story takes place in. The author defines a story space using our story representation, which is then passed to our storytelling mechanism, the director agent. The actors are semi-autonomous, intelligent agents. They have the capability to execute their own goal-based behavior or to base their behavior on commands from the director. The director's decisions in managing the story at any given time depend on the player's actions, the plot as specified by the author, the state of the world, and the director's projections of the player's future behavior.

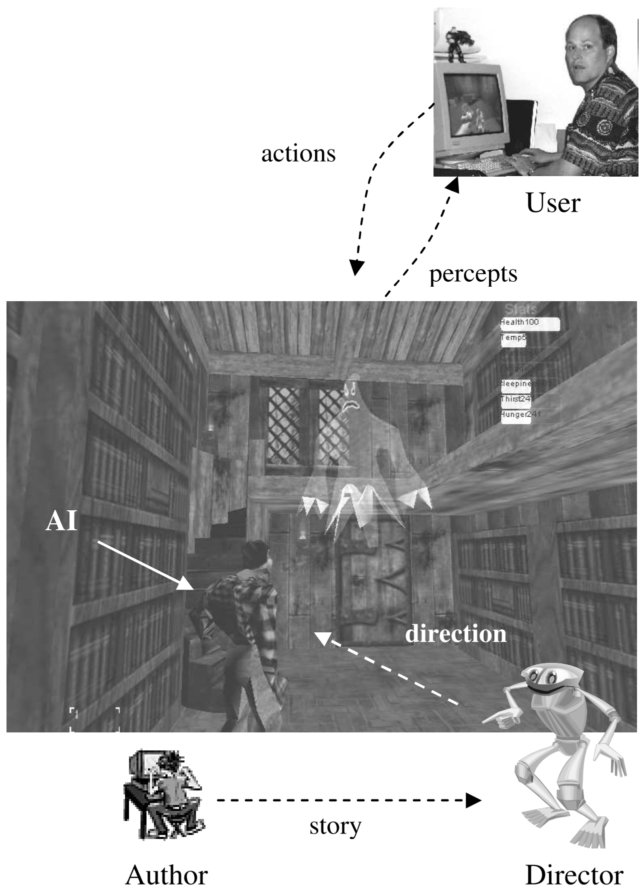

Figure 1. The Interactive Drama Architecture, IDA.

IDA's goal is to provide the means for a human author to specify plot content that describes a story space for the player to act in. The player should be able to execute actions that could possibly move him outside of that space, going against the plot description; unnecessarily constraining him goes against our goal of providing an interactive experience. This is the crux of the problem presented in MIMESIS (Young et al. 2004). The player can execute actions that threaten the preconditions of operators in the story plan. They respond to this type of action by either replanning and incorporating the action into the plan or by immediately disallowing the effects of the action altogether, such as by causing a bullet to always miss a character that is central to the ensuing plot. The purpose behind this philosophy is to allow the player to behave as he wishes. If the resulting behavior conflicts with the plot, then either the results of the player's action or the structure of the plot have to change.

IDA uses an omniscient story director agent to maintain the plot's progression and is in essence the "playwriting expert system" as described by Laurel above. Much like a human "dungeon master" directs a table top role-playing game, the director agent works with a pre-written story to guide the player through a story. The director follows the player as he progresses through the story, giving direction to characters when necessary to perform particular plot elements and to guide the player to stay within the story space. IDA's response to the problem of possible conflicting player behavior is one of *preemptive guidance*. IDA hypothesizes future player behavior, represented by the player's goals and the knowledge that they have gathered. If the system has a reasonable hypothesis of what the player will do in the future, it may use that hypothesis to subtly guide the player to stay within the story space, as opposed to solely rely on reacting to conflicting actions as they occur. This subtle guidance, used in combination with more immediately reactive guidance, should provide for a more coherent story experience for the player.

Our game environment, called *Haunt 2*, consists of a fully-structured story, synthetic characters that take part in the story, a 3-D world constructed with the Unreal Tournament engine (Magerko et al. 2004), and the story director agent. The story of *Haunt 2* involves the player playing the role of a ghost. He awakes to find himself in a bed and breakfast. He has no knowledge of how he got there, why he is a ghost when all the other characters are people, or what he should be doing. Only after he discovers his crumpled body, hidden in an unused room in the house, should he realize that he may have been murdered. It is up to the player to uncover who committed the murder, and then lead one of the others to his body, warning them that a murder has taken place. This paper defines the story representation that allows an author to represent *Haunt 2* in a cohesive manner with respect to the various system elements described above.

Story Representation Requirements for IDA

Below are the requirements for story representation in IDA. These requirements guide the design of the story representation language and form the basis of its evaluation.

1. Expressivity: The author should be able to express himself along a series of dimensions, including (but not necessarily restricted to) *dialogue*, *staging*, *character behavior*, *pacing*, and *environmental conditions* (e.g. lighting). The author should be able to describe how the characters act and present their dialogue, their blocking, and how the player fits into the story as a character. For example, an author for *Haunt 2* should be able to describe a scene introducing the Sally character to the player. He should be able to give her expository dialogue, describe where she should be relative to the player, what other characters could be part of this introductory conversation, and what state the player should be in.

2. Coherency: The author must be able to associate plot content with other content within the context of the overall narrative. This may be done implicitly or explicitly, as long as the space of possible stories that can be experienced by the player includes coherent, logically unfolding stories. A representation that did not meet this requirement would allow possible orderings of plot content that did not make sense in an overall narrative. For example, an introductory scene with Sally should not be allowed to occur after the concluding scene where the player leads one of the characters to discover the murder.

3. Variability: The story representation should not constrain the player's experience to a single possible narrative. It should support multiple paths through the space of stories defined by the plot content. The more possible orderings of plot content, whether explicitly or implicitly authored by the writer, the better the representation fulfills this requirement. When playing through *Haunt 2*, for example, there should be different narratives experienced when different decisions are made by the player

4. Supports Player Prediction: Player prediction can be a valuable asset to interactive storytelling. If the storytelling mechanism has an accurate hypothesis about what the player is trying to accomplish in both the present and in the future, then that mechanism can make a better-informed decision about how to manage the story's progression. The representation must, either implicitly or explicitly, allow the author to define a space of behavior that the predictive model can search through. A plot representation that generated future content would not necessarily provide a searchable future from a given point in the plot, thus making it impossible to compare a predictive model of player behavior with future content.

5. Fully-structured story: The author should be able to explicitly define a space of stories that carves out his artistic vision. Our goal is to create experiences in which complete dramatic stories can be authored, which is in contrast to systems that depend on a story, or at least the story's structure, to emerge. This requirement intends to provide a means for explicitly dictating when certain plot elements happen, either in relationship to each other or in reference to the overall passing of time in the narrative.

Story Representation

A story representation for an interactive drama is comprised of at least two main features, story content and structure, as described in requirements #1, #2, and #5. It is important for the author to have the means for describing what characters are involved in a scene, what they do, etc. It is also necessary to be able to describe how the story could have logically arrived at that scene and where it can go from there. Whether it is through rules that determine plot progression (Sgorous 1999), explicitly ordering plot content (Young et al. 2004; Magerko 2004), or defining a heuristic for future plot choice (Mateas and Stern 2002; Weyhrauch 1997), there should be some aspect of the representation that provides momentum to the story, that describes how it can progress.

Content

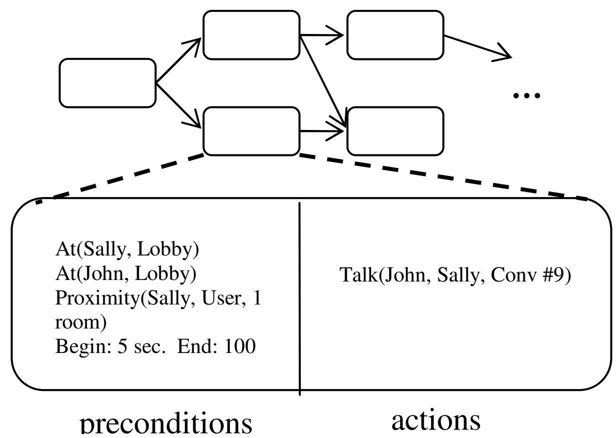

Figure 2. An example of partial-ordering of plot

The story representation used in IDA revolves around the partial ordering of abstract *plot points*, an example of which is shown in Figure 2. Each story event that takes place in *Haunt 2*, such as the player discovering his body or a scene revealing back-story between Sally and one of the other characters, is represented as a node in this graph. This graph structure, G, is represented as G → (N, E), where N is the set of nodes (or *plot points*) in the graph and E is the set of edges connecting them in a partial order. Plot points are defined as N → (P, A, c), where P is the set of preconditions for a node, which describes a set of world states where every p ϵ P is true, A is the set of actions for a node, which are the plot events that are performed after all members of P are fulfilled, and c is the timing constraint associated with this plot point, which describes a time span during which every p ϵ P must be true. A plot point's precondition, p, is a logical statement that describes what should be true in the world in order for the plot point's actions to be executed, similar to preconditions used in STRIPS-style representations. Our example plot point shown in Figure 2 illustrates some simple preconditions that are used in our story. The plot point's set of actions represents the performances that are made by the directable characters once the preconditions are fulfilled. This can be viewed as an explicit stimulus/response model of plot events; when the current state of the world is a member of a set of world states that meet a plot point's requirements, the corresponding actions are performed. As shown in Figure 2, if John and Sally are supposed to have a conversation in front of the player, the associated plot point's preconditions would be fulfilled, once everyone is in the same room, and the characters would be instructed to have a specific conversation.

At the beginning of the experience, any plot points without parents are labeled as *active*. The director keeps a list of all active plot points. Once an active point has been performed, it is removed from the list and any new points that are a) not on the open list and b) have no parents that have not been performed will be added to the list. Through these plot points, the author can specify the actual content of what happens, where it happens, and with whom. This relates back to requirement #1 listed above, which describes the need for expressivity in our representation along a series of dimensions. In Haunt 2, the preconditions that define a plot point may include the location of particular actors, their physiological and/or mental state (e.g. knowledge that they have), or their inventory. Actions are typically comprised of verbs, character performances that are dialogue, staging directions, or some desired change in the character's mental state, such as a new goal to achieve or fact to know.

The purpose of providing the author with timing constraints is to give him the means of specifying the *pacing* of the experience he is creating, which is another aspect of our representation that addresses requirement #1. Just as in other visual media, such as cinema (Field 1994), pacing is an important feature of performance. The timing constraint, c, associated with a plot point can be viewed as a special precondition for that plot point. As shown in Figure 2, a timing constraint has *begin* and *end* conditions, denoting the earliest time that the given plot point could be performed, and the latest possible time that the conditions could all be fulfilled. The director will not execute any director actions to fulfill a plot point's preconditions until that point's begin timing constraint has been fulfilled.

The *begin* constraint for a plot point marks the starting point for any directions to be given for that point. If there are preconditions that do not involve the player (e.g. *At(Sally, Lobby)*), the director will wait until the current time is past the *begin* constraint before directing the actors involved in that precondition. If all of the preconditions are fulfilled, then the director will wait until the current time is greater than the *begin* constraint before executing the plot point's action. The *end* constraint for a plot point represents an upper bound on how much time the player can be left to his own devices before the system decides to intervene. This defines the deadline by which the player and characters must fulfill the plot point's preconditions. If the constraint is violated while there are still unfulfilled preconditions left, the flow of the story, as specified by the author, has been violated and direction is needed to reconcile the current world state with the desired state. This use of pacing to help define the story space is a novel aspect of IDA's story representation.

By specifying timing constraints for plot points, the author is encoding how rapidly in succession events should take place in the story. For a quicker pacing, the author can make the interval between *start* and *end* relatively short and make the *start* constraint small, allowing the plot point to be performed quickly after its predecessor. This increases the amount of time between when the plot point becomes first active and when it can actually be performed. For slower pacing, the author can make the interval longer and the *start* constraint larger, creating an environment that allows the player more time to observe and interact with the environment. For example, in Haunt 2, the player is a ghost because his character was murdered by an unknown assailant at the beginning of the story. The rest of the plot involves the player figuring out who might have murdered him and helping one of the other characters find his hidden body to reveal the murder. As the player gets closer to reaching the main goal of helping another character uncover the body, I as an author would like the pacing of the story to quicken. I can achieve this by defining shorter spans of time with the latter plot points' timing constraints. If the player doesn't act quickly, the system will.

Structure

Plot points are connected to each other via directed edges (see Figure 2). These links do not represent paths in a story graph for the player to follow. They describe an explicit partial-ordering of the plot content. Therefore, as opposed to describing a graph for traversal, this representation describes a space of possible topological orderings.

In *Haunt 2*, there are plot points that we would definitely like to have happen before others. The end scene, concerning the player leading another character to the dead body, should happen after the player has learned about the seldom-used library and discovered the body there. The player could learn about this room in a number of different ways (e.g. from overhearing the Innkeeper mention it or by stumbling upon the room while exploring the building). However, there are other unrelated events, such as the player being initially introduced to the other characters, which can happen in a flexible ordering and would be structured with fewer ordering constraints than the ending.

This structure is similar to the planning language in MIMESIS described earlier. The key difference is that our representation has no explicit concept of causality. A STRIPS-style language contains both ordering links and causal links between plot operators, whereas our language can be viewed as an incomplete plan; it has no causal links. Our graph contains partially-ordered nodes, and those nodes do have preconditions. However, the nodes do not have explicit postconditions that can be causally linked to subsequent nodes, thus the absence of explicit causality.

This affects our ability to replan, like MIMESIS does, as an approach to story management. However, it offers a more streamlined representation that allows us to directly encode actor and director actions in the plot points rather than encoding the effects of those actions. If there is a particular behavior we want to see, it is represented in a plot point's actions rather than as a separate operator.

This representation forces the writer to consider the temporal flow of events when ordering them; it is possible to author a story that has dialogue in a plot point which refers to an event that hasn't occurred yet. There is no mechanical device, such as a planner, to ensure causality. Therefore, this representation partially fulfills requirement #2 by providing an explicit mechanism to describe temporal coherence, but not necessarily content coherence. It is quite possible for the author to write dialogue that refers to a past event (e.g. the Innkeeper warning the guests to leave the ghost they spoke of alone) that has not even occurred yet (e.g. the plot point with this dialogue does not explicitly come after the ghost being seen by any of the guests). This does provide the author with the means of creating large, "undefined spaces" in the player experience. If there is no explicit description of action and effect, there can be any sequence of events that can be combined to fulfill a plot point's preconditions (see requirement #3). Therefore, the plot does not represent an action-by-action account of what happens in the world; it is a skeletal representation that defines key points that the story should gravitate towards. How those plot points are fulfilled depends largely on the player's interactions with the world.

This explicit use of structure in our representation allows us to employ a predictive player model in IDA (see requirement #4). We can compare hypothesized player behavior against a well-defined story space. If the player's predicted actions move him outside of that story space, then the director can choose to preemptively direct the world, attempting to influence the player's behavior to avoid harming the story's progression. Without a story space to compare this prediction to, it would be much harder to take advantage of this useful story mediation tool.

Implicit Story Elements

As we stated earlier, the story graph represents a space of possible stories; the content defined by the author is the main influence on the narrative experience. However, there are other aspects of story that are less explicitly represented that we have yet to address. One implicit story element is the author's use of uninstantiated plot content. This is the ability to define story content in an abstract manner, allowing the definition of that content to be determined at runtime. The other implicit story element to consider is the contribution of director and character behavior to the narrative. While the authored content does define the story space for the player's experience, the "blanks" are filled in by the actions executed by the director, synthetic characters, and of course the player.

A simple, yet effective way for us to expand the possible size of a story space is to allow uninstantiated plot content. When the author does not want to write a specific detail, he can use a variable in a precondition or action that will be bound at runtime. What that variable can be bound to may be further constrained by the author. For example, the situation shown in Figure 2 may not necessarily involve John talking to Sally, but it just needs to be John talking to anyone aside from the player. Therefore, the author can simply replace *Sally* with a story variable (e.g. *At(x, Lobby)*), then constrain x != *Player*. This variable can be used in other plot points in the story, being globally defined once the first instance of it is bound. This provides the author with a mechanism of least-commitment authorship; he only has to be as specific as he desires. Important plot content should be tightly constrained, but the flexibility of the story representation allows the author to expand the size of the story space (see requirement #3).

Which actions are chosen by the director can also implicitly affect the narrative from experience to experience. The director's actions are a set of strategies for directing the characters (including the environment) to perform plot content or to encourage specific player behavior. These actions are categorized according to what kind of situations they are applicable to. Currently-implemented categories in Haunt 2 are: *location-actor*, *location-player*, *physiology*, *conversation*, *drink*, *proximity*, and *knowledge*. The actions in the *location-actor* category are strategies for getting an actor to a specific location, such as giving the target actor the goal of moving to that area or giving a second actor the goal of yelling "Hey, come over here…" to the target. Since there is more than one possible way for the director to fulfill a plot point's preconditions or perform its actions, this is another example of variability (see requirement #3) between experiences. The skeletal plot representation can be realized in different ways because the player or director makes different choices in an experience. Just as the strategy categories dictate which director actions are appropriate to fulfill a plot point's preconditions or to perform its actions, they also dictate which strategies are appropriate for guiding the player, either as a response to model failure or as a timing constraint violation.

By design, the characters that are involved in the story are not simple puppets that can only do the specific actions that they are given. We refer to our synthetic characters as *semi-autonomous* (Blumberg and Galyean 1997). This means that not only can they pursue their own goals when not involved in the plot, but they can also receive directions with varying degrees of specificity. If a character has no current directions to execute, it may be left to pursue its own goals until directed again. This gives the characters more believable behavior without burdening the director with determining their moment to moment behavior. The characters are agents developed in the Soar architecture (Laird et. al 1987). Soar includes long-term knowledge that supports a combination of reactive and hierarchical goal-driven behavior. Soar also includes short-term knowledge that maintains the agents' sensory information and situational awareness (Magerko et al. 2004). When an agent is sent a new command from the director, the command can be anything from a very high-level goal (e.g. "hang out anywhere") to a very specific one (e.g. "engage in conversation #113 with Sally"). These commands provide a method for the director to control the actors' behavior to carry out specific plot points specified by the author. How a character decides to fulfill a high-level goal or spend its time when pursuing its own goals adds to the variability of the player's experiences (see requirement #3).

Discussion

We have presented the requirements for our interactive drama architecture and how we have approached them. Requirement #1, expressivity, has been addressed by the first-order logic used by the representation. It is possible to specify plot content as well as pacing. A richer representation would also provide the means for such things as point of view, camera control (Jhala 2004), and ambient lighting (El-Nasr 2003), and how they relate to the plot. Requirement #2, coherency, is partially ensured by the temporal ordering of plot content. It is a means for the author to structure a coherent experience, but does, however, nothing to guarantee such coherency. Requirement #3, variability, has been met by various aspects of our representation, such as the variability inherent in our implicit plot content and the number of different possible total orderings allowed by a partial-ordering. Requirement #4, allowing player prediction, is clearly met by the representation. We have a mechanism for prediction and have used it in a story created with this representation. Requirement #5, allowing a fully-structured story, is also clearly met. Any temporal ordering of a set of plot points is allowed in the representation, ranging from a completely linear story to one that has no ordering at all.

In our experiences with authoring story content for *Haunt 2*, the obvious bottleneck for building a rich, interactive system is the large amount of content needed; the story we tell in our system is just a bare minimum of content to explore the architecture's capabilities. While our story representation in IDA has given us the opportunity to explore some interesting approaches to story direction, like the use of a predictive model and uninstantiated plot content, it has not made it any easier to author an interactive drama compared to other systems. A significant improvement in writing content would be made by developing tools that facilitated the authoring process.

References

Blumberg, B. and Galyean T. 1995. Multi-level Control for Animated Autonomous Agents: Do the Right Thing...Oh, Not that. In *Workshop Notes of the 95 AAAI Symposium on Interactive Story Systems*.

Cavazza, M., Charles, F., and Mead, S. 2002. Sex, Lies, and Video Games: an Interactive Storytelling Prototype. In Proceedings of *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*.

El-Nasr, M. S. and I. Horswill. 2003. Real-time Lighting Design for Interactive Narrative. In *Proceedings of 2nd International Conference on Virtual Storytelling*.

Gordon, A. and Iuppa, N. 2003. Experience Management Using Using Storyline Adaptation Strategies. In *Proceedings of First International Conference for Technologies for Interactive Digital Storytelling and Entertainment*.

Jhala, A. 2004. *Intelligent Cinematic Camera Control for Dynamic Narratives*. Master's Thesis, Department of Computer Science, North Carolina State University.

Laird, J. E., A. Newell, Rosenbloom, P.S. 1987. *Soar: An architecture for general intelligence*. Artificial Intelligence 33(3), 1-64.

Laurel, B. 1986. Toward the Design of a Computer-Based Interactive Fantasy System. PhD diss., Drama Department, Ohio State University.

Magerko, B., Laird, J. E., Assanie, M., Kerfoot, A., Stokes, D. 2004. AI Characters and Directors for Interactive Computer Games. *16th Innovative Applications of Artificial Intelligence Conference*, 877-883. San Jose, CA,

Magerko, B. and Laird, J. E. 2004. Mediating the Tension Between Plot and Interaction. In *Proceedings of AAAI Workshop Series: Challenges in Game Artificial Intelligence*, 108-112. San Jose, CA.

Mateas, M. and A. Stern. 2002. A Behavior Language for Story-Based Believable Agents. *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*.

Sgouros, N. M. 1999. Dynamic Generation, Management, and Resolution of Interactive Plots. *Artificial Intelligence* 107: 29-62.

Szilas, N., Marty, O., Réty, J. 2003. Authoring Highly Generative Interactive Drama. In *Proceedings of 2nd International Conference on Virtual Storytelling*.

Field, S. 1994. *Screenplay: The Foundations of Screenwriting*. New York: Dell Publishing Company.

Weyhrauch, P. 1997. *Guiding Interactive Drama*. PhD Thesis, Carnegie Mellon.

Young, R. M., Riedl, M. O., Branly, M. Jhala, A., Martin, R.J., Saretto, C.J. An Architecture for Integrating Plan-based Behavior Generation with Interactive Game Environments. *Journal of Game Development* 1(1): 51-70, 2004.

Acknowledgments

Special thanks to the Institute of Creative Technologies for funding this research and to the Soar Games Group for their work on *Haunt 2*.