

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228378172>

Evaluating preemptive story direction in the interactive drama architecture

Article · January 2006

CITATIONS

30

READS

158

1 author:



[Brian Magerko](#)

Georgia Institute of Technology

102 PUBLICATIONS 1,051 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Drawing Apprentice: Co-Creative Drawing Partner [View project](#)



Quantifying Human Interaction Dynamics in Open-Ended Improvisational Creative Collaboration [View project](#)

All content following this page was uploaded by [Brian Magerko](#) on 23 April 2015.

The user has requested enhancement of the downloaded file.

Evaluating Preemptive Story Direction in the Interactive Drama Architecture

Brian Magerko, Ph.D.
Games for Entertainment and Learning Lab
Michigan State University
417 Communication Arts Bldg., East Lansing, MI 48823
<http://gel.msu.edu>
magerko@msu.edu

1. Introduction

In an author-centric interactive drama, both the player's decisions and the author's desires should coherently influence the player's individual story experience. Different player interactions with the system should yield different stories, just as different authored content would. By defining experiences covered by the authored content (the set of which I call a *story space*), the author is creating an artistic vision for the player to take part in. As opposed to having explicit choices for the player to choose from and constraining those choices, interactive drama attempts to offer the player a fluid, continuous dramatic experience, akin to taking part in an improvisational play where the player is the protagonist in the story (Kelso et al. 1993; Laurel 1986).

The problem with relying on a playwriting expert system (e.g. a story director agent) such as Laurel proposed in her seminal dissertation thesis (Laurel 1986) is that it becomes difficult to constrain the interactivity so that the player does not perform actions that take the experience outside of the content defined by the story space (i.e. the player executes an action that should have a dramatic consequence, such as killing a main character, but no content is written to cover the situation). This is a key problem when attempting to explore more relaxed approaches to interactive drama rather than relying on the traditional use of story graphs or branching storylines (Fahlstein 2005). More player choices lead to more opportunities for him to harm the progression

of the plot. I define this issue of player actions bringing a dramatic experience outside of the boundaries of authored content in an interactive drama as the *boundary problem*.

Research in the field of interactive drama investigates methods for both providing the player with a greater sense of interactivity and addressing the boundary problem described above. These methods include natural language understanding and generation, which provides the player with a powerful means of interacting with synthetic characters and avoids the boundary problem by using catch-all phrases in response to ambiguous player inputs (Cavazza 2005; Mateas and Stern 2003; Zubek 2005); new story representations, which provide the means for creating larger story spaces (Fairclough 2004; Mateas and Stern 2002; Young et al. 2004); the creation of synthetic characters (Blumberg and Galyean 1997; Gratch et al. 2002; Loyall 1997; Reilly 1996); automated story management (Fairclough 2004; Gordon and Iuppa 2003; Magerko and Laird 2004; Weyhrauch 1997; Young et al. 2004); generating dramatic situations (Bringsjord and Ferrucci 2000; Fairclough 2004; Meehan 1981; Sgorous 1999); and developing methods for the incorporation or prevention of player actions that cause a boundary problem (Young et al. 2004).

These approaches either rely on a) strategies that mainly apply to boundary problems in discourse as opposed to physical boundary problems (*discourse*), b) assumptions that many different dramatic situations can be reliably generated by rules (*generation*), or c) strategies that disallow player actions from having logical or expected consequences (*prevention*). They do not provide a means for *subtly* addressing the boundary issues that may arise in an interactive drama that arise from *physical interactions* with the world, such as attempting to kill a character that is key to future plot development.

My approach to subtly address the boundary problem in the physical domain is to use an omniscient story director agent, implemented in the Interactive Drama Architecture (IDA),

which uses a predictive model of player behavior to maintain the plot progression. Much like a human dungeon master does in some table top role-playing games, the director agent works with a pre-written story structure and attempts to guide the player through that story. The director follows along with the plot as it moves along, giving commands (called *directions* or *director actions*) to characters when necessary to perform particular plot elements. As opposed to waiting until boundary problems occur, the director agent attempts to predict the player's future behavior so that it can preemptively, though subtly, steer the player away from actions that may endanger the progression of the plot. *Haunt 2*, which is the game environment used by IDA that has been developed with the Soar Games group at the University of Michigan, 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, which is the focus of this research.

The ability to preemptively direct is what distinguishes IDA most from other interactive drama systems, such as the MIMESIS architecture and the Crosstalk framework (Klesen et al. 2003; Young et al. 2004). MIMESIS uses a fully structured plot, represented as a partial-order plan, and either incorporates unplanned player actions into the story or avoids them altogether if incorporating them is infeasible. The CrossTalk framework incorporates plan-based automatic dialogue generation with an author-defined narrative graph. Other approaches to interactive drama have taken a more modular approach to plot construction so that there is no single coherent plot that is explicitly created by the author (Mateas and Stern 2002; Sgorous 1999; Weyhrauch 1997). They rely on heuristically choosing plot elements as the player moves through the space of possible stories. Some systems have also included a player history as a model of player experience to help heuristically choose what plot elements should occur next (Szilas et al.

2003; Weyhrauch 1997). What these systems do not address is how to avoid problematic player actions (i.e. boundary problems) *before* they occur by employing some prediction of future events (Beal et al. 2002). IDA makes use of prediction to subtly guide the player's behavior to stay within the bounds of the authored story content and looks at methods for expanding the number of possible player experiences and thus the size of potential story spaces.

2. Related Work

This section examines several different fields of computational research related to this work: automated storytelling systems, commercial computer games, and interactive drama. Automated storytelling systems are the artificial intelligence approach to generating complete stories from a set of knowledge. Commercial computer games involve approaches to using the computer as a medium for players to interact with a virtual story world.

2.1. Automated Storytelling

Automated storytelling is the direct digital predecessor of interactive drama. This field has focused on creating programs that use pre-authored knowledge and logical representations of narrative structure to create new stories without any real-time input from a player. The main difference between automated storytelling and interactive drama is the lack of interaction in automated storytelling. The “player” simply reads or watches what an automated storyteller produces, rather than actively taking part in the experience. However, these systems deal with problems commonly found in interactive drama, such as plot representation, character behavior, and story structure.

AI systems such as TALE-SPIN, UNIVERSE, MINSTREL, and BRUTUS are expressive in terms of the kinds of stories that they can tell. The first major system, TALE-SPIN, introduced

narrative as a planning representation (Meehan 1981). Story content in TALE-SPIN is represented as character goals and operators that can be used to achieve those goals. Operators may represent a set of subgoals or an atomic action that changes the world state. Leibowitz's UNIVERSE expounded on this original design to define plot elements not as the goals and plans of the characters but rather as those of the *author*, which may be a human or program. As is typical in story generation algorithms, neither include the player as a character in the story.

The MINSTREL system decomposes this notion of explicit authorial goals into *thematic goals*, *consistency goals*, *drama goals*, and *presentation goals* (Turner 1994). Thematic goals center the selection of story content on a chosen theme (e.g., "history repeats itself"). Consistency goals ensure that a story is plausible and believable. Drama goals employ dramatic writing techniques to make the story more enjoyable. Presentation goals make the presentation of the story a pleasurable one for the reader. This decomposition still does not allow for the creation of detailed narratives; Turner's theory of story generation ignores concepts like character development, internal dialogue, rich environment descriptions, etc.

A final story generation system, BRUTUS, relies on a rigorous definition of dramatic principles, such as "betrayal" or "heartbreak," to generate stories that reflect those principles (Bringsjord and Ferrucci 2000). The authored initial components for the story generation are two large bodies of knowledge: *domain knowledge*, a formal description of the story domain, including objects, attributes, relationships, goals, and events; and *literary knowledge*, the set of principles that are used to encapsulate "telling a good story," which includes using *thematic knowledge* to engage the reader in classic themes and *story grammars* to instantiate classic story structures. *Thematic knowledge* is a formal definition of a thematic concept, such as "betrayal," allowing the author to guide the generation of stories to any defined literary theme. A *story*

grammar is a formal grammar that describes how a story can be recursively decomposed until atomic statements are reached (e.g. Story → Setting + Theme + Plot + Resolution, Setting → Characters + Location + Time, etc.). This gives the author the choice of generating story according to a specified structure. While BRUTUS does not provide the means for player interaction, the space of possible stories is increased due to BRUTUS' focus on creating different stories that may come from the same thematic material. However, a large investment in knowledge engineering is needed (e.g. domain knowledge-base, character definitions, rigorous definition of thematic concepts, etc.), which the authors admit is incredibly time-intensive. Overall, BRUTUS illustrates the means to center computer-generated drama on core literary themes, and also highlights the difficulty in proceeding with this process from a knowledge-engineering level for a single theme.

These systems are based around the creation of a standard representation that can be used to describe characters and their goals, the story's setting, etc. In other words, these systems are provided with the elements of a story so that a complete story can be created. They have complete control over their characters, unlike interactive drama which have human characters typically involved as the protagonist. Approaches used in automated storytelling, aside from the direct control of synthetic characters, do not directly transfer to interactive drama unless the player's actions are constrained down to the specifics in the created story, which then loses any sense of interactivity. However, these systems do supply a standard representation for authoring narrative systems and make suggestions for how generative techniques could be used in interactive drama.

2.2. Commercial Computer Games

Commercial computer games are a highly interactive form of entertainment that often incorporates storytelling into gameplay. Even in the early history of the industry, *interactive fiction* games were attempting to involve the player in a story as the main character. Interactive fictions, such as *Zork* (1980) or the infamous *Hitchhiker's Guide to the Galaxy* game (1984), are text-based experiences where the main input to the player is a rich textual description of the environment, items nearby available to him, etc. The player can manipulate objects in the world, interact with scripted characters, and navigate through a very discrete physical space (i.e. maps or “dungeons” are typically organized as directed graphs, with rooms as nodes and the connections allowed between them, such as entryways or trapdoors, as directed edges), but the player's actions ultimately had only limited impact on the evolution of the story.

Storytelling in computer games has been involved in each new generation of games. Games that typically include some aspects of storytelling are interactive fiction games and adventure games (first person shooters and role-playing games have increasingly focused on story as well). Approaches to interactive fiction heavily influenced the design of graphical-based adventures games, such as the Sierra Games *King's Quest* series (REF) or more contemporary adventure games like *Grim Fandango* (1998), *Syberia* (2002), or *The Indigo Prophecy* (2005).

Interactive fiction games are text-based games that provide the player with a virtual world that is described by pre-authored text. The player can typically execute commands from a command prompt (e.g. “flip switch” or “north” to move north) in a turn-based fashion to move around and affect the world. These types of games offer a more specific dramatic character for the player to assume, which is contrary to the gung-ho, transparent characters that are used in FPSs. The gameplay is more centered on a narrative experience and less around combat. They

initially offered rich game text-based experiences at a time when computers had little graphics capabilities; the environment is all described with static text. Popular titles like *The Hitchhiker's Guide to the Galaxy*, *Planetfall* (1983), and *Zork* offer a first-person dramatic experience that involved the player examining his surroundings, solving puzzles, and interacting with fairly shallow synthetic characters all through simple text commands.

Adventure games, like the *Myst* series (1993) or *Gabriel Knight 3* (1999), are the graphical descendants of early interactive fiction. These types of games often offer a first-person experience in a three-dimensional world. However, these games focus much more on problem-solving and moving through a narrative structure with the player typically as the protagonist in a mystery story.

Interactive fiction and adventure games offer a narrative experience without any large degree of interactivity. For instance, if a player is playing through an interactive fiction and does not behave exactly how the story describes, then he will effectively be “stuck” in the plot. The burden is put on the player to figure out what is expected of him in order to advance the story. A more interactive, author-centric system would encourage the player to advance the plot if it noticed that his behavior was lagging with what was desired in the narrative.

The replay value of these systems has been a major issue in game design. They have yet to reach the point of offering significantly different narratives due to different player behaviors. This has been largely due to the lack of a significant amount of available consequential player actions and corresponding authored content. This issue is related to the choice of story representation used in these games. Story is typically represented as a story graph, where the player moves from one state to the next after he has found a new clue, solved a puzzle, or executed some misstep. As discussed in Section 1, story graphs are a representational dead-end

to authoring interactive drama. Although modern interactive fiction strives to push on this limitation, the benefits of a good interactive fiction experience are mainly from the superb authorship of different plot lines and the smooth integration of puzzles. This is the largest difference between these genres and interactive drama; the focus on dramatic interactivity suffers in these genres.

2.3. *Interactive Drama*

The computational approaches to interactive drama strive to create a new narrative medium that involves the player as a main character in a play that responds to his choices in the storyworld. The field of contemporary interactive drama research can be roughly divided into three different approaches: agent-based, director-based, and generative approaches. Agent-based systems, such as in Cavazza's systems (Cavazza 2005; Cavazza et al. 2002) and Blumberg's work (Blumberg and Galyean 1995; Blumberg and Galyean 1997), focus on strongly autonomous characters to elicit an emergent narrative experience. For example, Cavazza's Interactive Storytelling system involves the player not as a main character in the story, but as a "trickster" character that can alter the world to force changes in character behaviors (Cavazza et al. 2002). Roles are defined for each synthetic character in the forms of plans, which are broken down as hierarchical task decompositions. For instance, for one character to accomplish his goal of getting a date with a female character, he must fulfill several subgoals, which in turn have subgoals associated with accomplishing them, etc. The player can directly affect the characters by either manipulating physical objects that are on stage or by advising the characters via speech recognition. This approach to interactive drama does offer interactivity with a story, but in a very peripheral manner since the player is not actually *in* the story. Agent-based approaches in

general have issues with story being represented as *character goals* as opposed to higher-level *story goals* that may conflict with autonomous behavior (Mateas and Stern 2000).

Relying on the use of an intelligent “story director” (or “drama manager”) has emerged as a common design for coordinating the behavior of non-player characters and the player’s behavior in relation to pre-authored story goals (as opposed to character goals, discussed above). The MOE agent from Carnegie Mellon’s Oz Project is the part of the Oz architecture that focuses on search-based story direction (Weyhrauch 1997). The goal of MOE’s design was to build an automated story director that helped guide the player through a dramatic experience. MOE can affect the player’s behavior by executing actions in the world (called MOE-MOVES). It relied on an adversarial search algorithm to consider the possible complete stories in order to inform its decision about how to guide the player’s behavior.

MOE’s approach to interaction is to provide a connection between player actions, authorial desires (via the author-defined heuristic for search), and the temporal ordering of content. While MOE was a good step towards exploring interactivity in a dramatic context, it lacks any connection between scene *content* and player behavior. With MOE, the ordering of the scenes may vary as well as the MOE-MOVES used, but the content of each scene is fully specified beforehand. As Nelson and Mateas have discovered, the results of this approach do not appear to generalize across different stories (Nelson and Mateas 2005).

Another director-based approach, Façade is the first fully implemented interactive drama system (Adams 2005; Mateas and Stern 2003). Façade’s goal is to offer a complete, real-time dramatic experience with a highly interactive, character-driven story for the player to play a key role in. Façade avoids describing narratives with an explicit branching structure and instead introduces a representation language called ABL to specify both plot and character behavior in

an intertwined fashion. The smallest narrative units used are "beats," which are selected in succession as the story progresses.

Façade attempts to achieve a coherent experience via the team coordination of the synthetic actors. Within each beat, a joint behavior is specified that tells each character how they may behave, much like the multi-agent coordination work done by Tambe on STEAM (Tambe and Zhang 1998). A director agent is used to coordinate the behavior of the synthetic characters which, as mentioned earlier, are weakly autonomous characters.

This approach offers one of the more promising frameworks for an interactive drama. It provides a method for coordinating character behavior, animation and dialogue as a dramatically-motivated response to character input. However, the focus on interaction in Façade is primarily in the conversational realm; physical interactions are mainly incidental. This is very different from making a tangible, observable change to the physical environment that logically keeps the plot from progressing. A physical change to the world, such as attempting to shoot an important character, can bring the story to a complete halt (e.g. Imagine Frodo losing the One Ring at the beginning of *The Lord of the Rings*) compared to an odd interjection into a conversation, which may be handled awkwardly yet will still allow the story to logically continue. Façade tackles some difficult problems in interactive drama not addressed in this dissertation, but does not address the main difficulty, the boundary problem, that IDA does.

Generative systems are inspired heavily by the story generation community. A typical approach in these systems is to logically represent dramatic situations and recognize opportunities to have those experiences occur. DEFACTO approaches interactive storytelling by relying on logical definitions of dramatic concepts to guide the story, similar to the representations used in BRUTUS (Bringsjord and Ferrucci 2000; Sgorous 1999). The initial

inputs to the system describe initial plot conditions, role descriptions for characters, and rules on how the social world in the story acts. As the player interacts with the world, the director agent continuously executes a decision-making cycle of *generation*, *evaluation*, and *resolution*. This cycle involves the proposal of possible character interventions in the world, which are different character actions that would have an effect on the player's experience. Rules are defined in DEFACTO to evaluate these proposals based on their dramatic merit. An example dramatic situation encoded into DEFACTO is the *lifeline*. The *lifeline* rule describes an “unfavorable event” occurring by the hands of a non-player character followed by a “favorable event.” A suitable character intervention is chosen and then enacted in the world, leading to the beginning of the cycle again.

Both DEFACTO and OPIATE (Fairclough 2004), a related system that uses logical definitions of Proppian roles and events for generation, are examples of approaches to interactive drama that involve no directly authored plot constructed beforehand. Instead, the author of the experience defines such things as character roles, social constructs, and dramatic principles in order to indirectly specify the player's experience. While this possibly provides a more interactive experience, it is a likely bottleneck on artistic expression. Specifying dramatic rules so they are both general enough to cover many similar situations, but specific enough to have any real meaning, is precisely the difficulty pointed out by Bringsjord and Ferrucci (Bringsjord and Ferrucci 2000).

Another approach to story generation has been explored in the closely-related IN-TALE and MIMESIS projects (Riedl and Stern 2006; Young et al. 2004). MIMESIS, and IN-TALE in subsequent work, is an approach that specifically focuses on how to approach the boundary problem using replanning and story mediation techniques. Story content is represented as a plan,

authored by a human writer in a STRIPS-style planning language. The plan is comprised of actions executed by the player as well as the synthetic characters. As the player is executing an action, if the effects of that action will cause a threat (i.e. a boundary problem), the system relies on two mediation approaches to keep the story progressing. The first, called *accommodation*, quickly tries to replan, incorporating this new player action into the existing set of operators so the plan's goal may still be achieved. If replanning does not yield a new, usable plan, then the director will execute an *intervention*. Intervention involves executing some director action in the world (similar to how MOE affects to the world) to prevent the effects of the player's actions. For example, if the player tries to shoot a main character in the plot, and accommodation is not successful, then the director will make the player's shot miss, or perhaps make the gun jam. The director relies on strategies that deal with problems as they arise, an approach I call *reactive direction*, to address the boundary problem.

MIMESIS is the primary example in these architectures that both provide for a fully-structured authored plot *and* a mechanism for dealing with the boundary problem. The reliance on replanning is a seamless approach to story generation to accommodate player actions. However, when reactive strategies are frequently relied on to deal with more problematic situations, believability suffers. The strategies available to the director for reactive direction are limited to methods that can only be used as an immediate response to the boundary problem. Methods that are used when a boundary problem is anticipated, such as having the character in question yell "Don't shoot! I have something to say," are not possible under the guise of reactive direction. This approach to story direction has heavily influenced the approach to the boundary problem discussed in this paper.

3. IDA

My approach to address the boundary problem is the *Interactive Drama Architecture* (IDA). As shown in Figure 1, the architecture is comprised of an author, who writes story content, the director agent, the synthetic characters, the environment, and the player. IDA's central mechanism is the director, an agent that is responsible for managing the story in relation to the synthetic characters' behaviors and the player's interactions with the world.

3.1. *Haunt 2*

The story world that I have built in collaboration with the Soar Games Group is called *Haunt 2*, which is built using the Unreal Tournament game engine (Magerko et al. 2004). The story of *Haunt 2* puts the player in the role of a ghost. The player wakes up in a bed and breakfast, not knowing how he got there or why he is a ghost. The building is populated by three other characters: the Innkeeper, the man who runs the bed and breakfast; John, a professor staying at the Inn; and Sally, an ex-girlfriend of the Innkeeper's who is visiting the inn. The story, which is a very simple one, unfolds as the player discovers his dead body, leaving it up to him to try and figure out not only who murdered him, but how to encourage the innocent character to come across his body and alert the authorities to the foul deed.

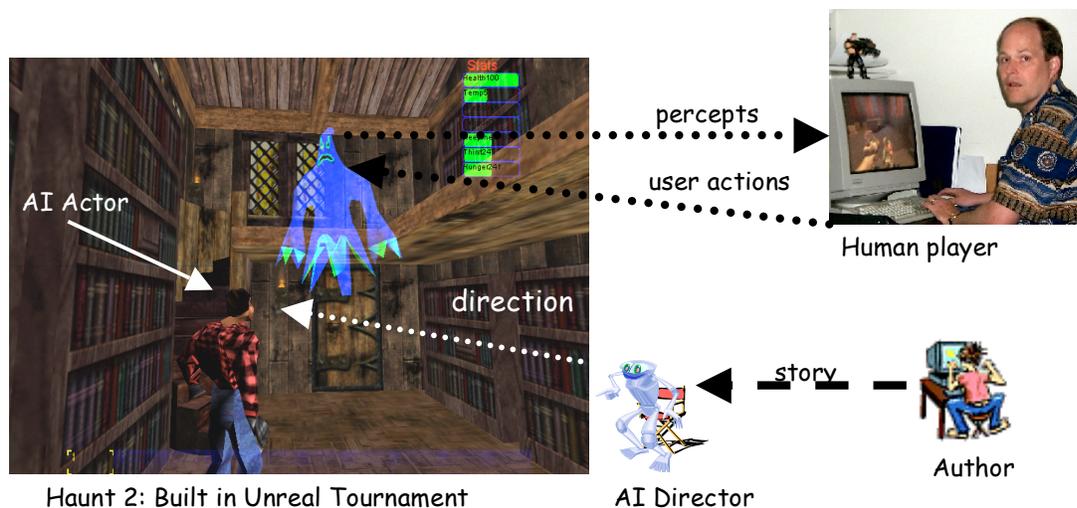


Figure 1. The Interactive Drama Architecture (IDA).

Each AI character is implemented as rules in the Soar architecture (Newell 1990). The characters share the same basic knowledge base to support interacting with the world and other characters. Specific characters have different physiologies, goals and background knowledge. For a thorough discussion of the integration of *Haunt 2* and IDA, see Magerko, et al. (2004).

3.2. Author

One of IDA's functions is to serve as a communicative tool for the author's artistic vision. The author uses the story representation provided by the architecture to sculpt a story space, which defines the possible stories that the player can experience. This space is defined both by the authored story content (e.g. *who*, *what*, *how*, and *where*) and by the story structure (i.e. *when* events happen in relation to other events). A brief description of the story representation used is given in Section 3.4.

The author acts as both artist and programmer, filling in all domain-dependent content for an interactive drama. Aside from the story description given to the director, the author is also

responsible for specifying any domain-dependent functions of the director, the environment and art content, as well as the synthetic character behaviors. The author, for my purposes, is viewed as a single person, but can easily be a conglomerate of artists and / or programmers.

3.3. Synthetic Characters

The synthetic characters in the world exhibit rational goal-based behavior. The agents for *Haunt 2* (called *HauntBots*) are authored in Soar and are defined by their long-term knowledge, which is a hierarchical set of goals, short-term knowledge, which are any working memory elements (WMEs) that are constructed from observing and interacting with the world, and preferences, which are relational control knowledge that probabilistically affect how operators are selected. The characters have basic world knowledge, such as how to navigate around the building, how to use objects (e.g. a match can be lit and a thermos can be drunk from), and how to hold a conversation with other characters. As mentioned above, they can connect to the *Haunt 2* environment via either sockets or multiple Soar agents can be directly embedded within the UnrealTournament process using a C-based API.

The primary drive for autonomous character behaviors is the Soar decision cycle of processing *input*, *proposing operators* that match on working memory, *selecting* one of those operators, then *applying* that operator and producing an *output*. The synthetic characters are capable of exploring the world, using matches to start a fire to warm themselves, and eating or drinking available food or water. The agents used in IDA are also *directable* by design. While they may be allowed to act on their own internal (i.e. cognitive and physiological) drives, it is much more common for them to receive commands from the director agent, telling them how they should behave in relation to the plot. A command can be any goal in the agent's hierarchy. The agent can be given a top-level goal, such as "explore," or very specific low-level actions,

such as “perform dialogue #131 with John in the library and then run away to another room.”

An agent takes this input as its new current goal. Once that goal is fulfilled, it moves on to the next commanded goal or goes back to relying on its own goals.

If the author defines plot content that is at a high / low / intermediary level in terms of character behavior, the synthetic characters can take that content as a command and execute it. One issue it does not address is the believability of the characters taking direction. The agents may be directable, but they instantaneously take on their new goal as opposed to smoothly transitioning from whatever they were doing to this new and possibly completely unrelated behavior (Assanie 2002). For example, if Sally and John are in the middle of a conversation, directing Sally to immediately break off from that conversation and go speak with the Innkeeper may be awkward at best. In terms of believability, it would be much better if Sally received that command and then pardoned herself from her current conversation before moving to another. This work does not specifically focus on the design and believability of synthetic character behaviors (as opposed to the believable management and coordination of story events), so this instantaneous goal transition is an acceptable, though less than ideal, approach. The author, who has several artistic and programming duties that may involve more than one actual person, is involved in the agent construction process. Therefore, the author does have knowledge about how an abstract behavior might be elaborated.

3.4. Story Representation in IDA

IDA’s story representation uses a partial ordering of abstract *plot points*. This graph structure, G , is represented as $G \rightarrow (N, E)$, 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 $N \rightarrow (P, M, A, c)$, where P is the set of preconditions for a node, which describes a set of world

states where every $p \in P$ is true, M is the name of the plot point, 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 $p \in P$ must be true in the story world. For a more thorough description of IDA's story representation, see Magerko and Laird (2005).

3.5. Story Director

IDA's director agent responds to the boundary problem by attempting to influence the player's behavior (through *story direction*) to keep the experience within the story world. The director's decision to guide the player's behavior is dependent on both the player's current actions in the world as well as the director's hypothesis of future player actions. Systems like MIMESIS or ALT-SIM have a similar view of this problem: the player can execute actions that are in conflict with story content...*what can the system do to deal with this?* Their approaches focus on responding to player actions as they are being executed. MIMESIS relies on replanning to incorporate an unexpected action into the story plan. If replanning fails, then the effects of that action are disallowed (e.g. forcibly making the player miss when attempting to shoot a critical character with a gun). ALT-SIM attempts to employ guidance strategies that push the player towards a particular event (e.g. drawing the player's attention to an important coffee shop he has walked by and paid little attention to), but doesn't necessarily deal with potentially critical problems, like shooting a main character as MIMESIS does (Gordon and Iuppa 2003). IDA employs *reactive direction* to alter the story or story world to address problematic player actions.

The problem with relying on reactive direction is that it puts a premium on the effectiveness of story direction as opposed to preferring more subtle strategies. Once the player attempts to execute a problematic action that creates a boundary problem there are only so many strategies

that can be used. Moreover, the repetition of these strategies can have an extremely negative effect on believability. For example, you only get “one coincidence” in a story according to screenwriting theory (Field 1994). One coincidence, such as a character miraculously surviving a fall from a building by landing in a cushy, trash-filled dumpster, is typically acceptable for an audience. Any more than that will likely make the story seem contrived and unbelievable. Therefore, even if a reactive strategy is fairly believable (e.g. missing the targeted character from fifteen yards away), the repeated use of these strategies harms the perceived plausibility of the story experience.

IDA addresses this problem by attempting to predict the player’s behavior to affect when and how it attempts story guidance, which is a major contribution of this work. If the hypothetical future player behavior indicates that a boundary problem is likely to occur, then the director employs story guidance strategies that are subtler than strategies that apply to when a problem occurs (which I define as *preemptive direction*). One could construct an agent that *always* used these subtle strategies whenever possible, but a design that relied on the director to execute actions all the time would harm believability. The hypothesis is that the preemptive approach will need to execute *fewer* directions. Its design is to both use subtle story direction when possible, but only to use it when deemed necessary by the predictive model. This design provides the intelligent use of subtle strategies without having to rely on executing them at every possible moment.

Consider an interactive drama that involves the player in a road trip story like in the film *Sideways* (Payne 2004). The player is to travel with his buddy from San Francisco to Sonoma Valley and craziness ensues once they get there. Suppose that the player has chosen to drive there via Highway 1, a road that has numerous stretches of dangerous sharp curves bordered by

the ocean on one side and a mountain wall on the other. The director agent for this system observes them driving much too fast and recklessly down the highway. It is obvious that an accident could easily occur, which would disrupt the flow of the story experience. If the director only uses reactive strategies, then it will passively observe the scene, waiting for an accident to happen. When the player *does* hit a curve too sharp and goes careening off of the cliff, the director observes that the story is definitely disrupted and does something, such as throwing the player and his buddy from the car at the last second to keep the story going. As pointed out earlier, this kind of effective reactive strategy may work once, but the plausibility of the story world is definitely harmed if such a coincidence happens again in the future (the fact that they survived this single accident at all could even put a strain on the plausibility of the story world).

A director that uses preemptive direction would observe the scene and predict that there is a high likelihood of an accident. The director would attempt to alter the world to preemptively discourage that problem from occurring in a *subtle, believable fashion*. The system could spawn a cop behind the player and have the cop pull him over, spawn a truck in front of him around a curve so he has to slow down to the truck's speed, or even have a fuel injector blow in the car's engine, greatly harming the car's capabilities. These strategies, which could not be used for reactive direction, can fit into the game world in a realistic manner (e.g. the player can get pulled over, then later encounter a slow-moving truck that that is blocking both lanes) without straining the dramatic plausibility of the story world. The player's behavior can be subtly guided in this manner, while relying on reactive direction as a last resort (e.g. the player tries to swerve around the truck and goes flying off the road).

4. Player Modeling

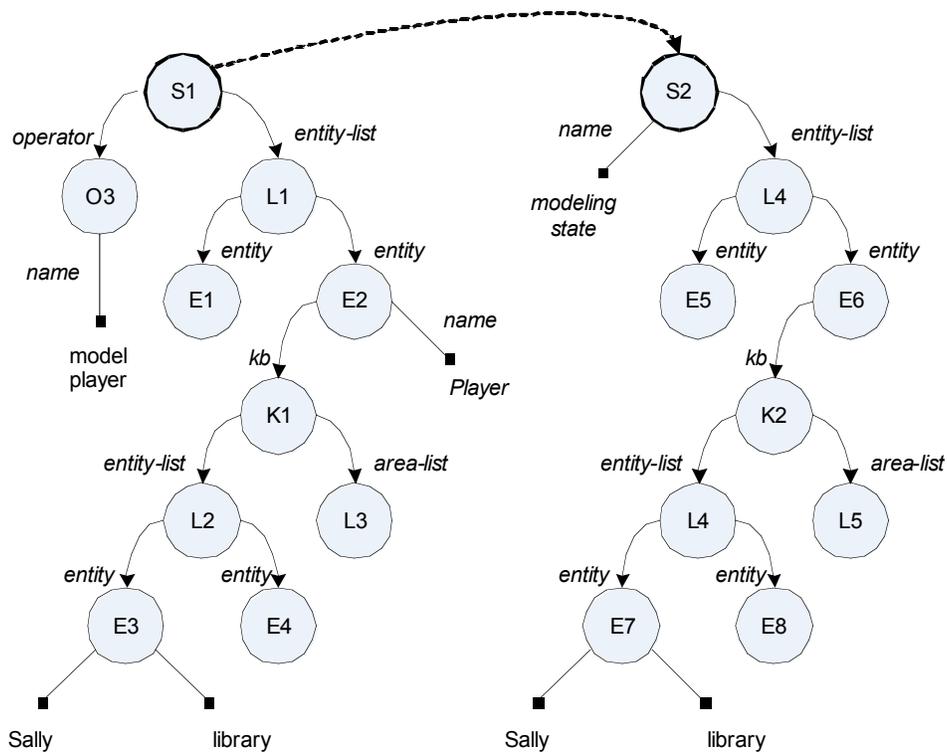


Figure 2. Example of copying information, including hypothesized player knowledge, to the modeling state for prediction.

IDA is designed to avoid conflicting player behavior before it occurs. The system models short-term player behavior and treats the results of that model as a hypothesis of player behavior in the near future. If the player is hypothesized to create a boundary problem, then the director will try to influence the player's behavior to avoid that possible future. This hypothesis involves the director a) creating an internal simulation of the game environment, b) running an author-defined player model on that environment, and then c) building a hypothesis of the probability of future boundary problems by observing the model's behavior and comparing it to the plot representation (Laird 2001). This process gives the director a hypothesis about what may happen in the future as well as the probability of that future occurring. Whenever a plot point is finished,

the director begins modeling by creating an internal copy of the world state, which includes the director's hypothesis of the player's knowledge base. This internal copy is treated as a virtual representation of the environment that the model can reason about and execute actions in. It is a working memory structure identical to the director's current representation of the world state. Figure 2 illustrates a subset of the information that is copied over during prediction. The identifiers, such as "S1" or "E8", are unique identifiers in working memory. When a new state is created for modeling (S2), it contains new WMEs that are created for modeling. The structure of working memory and the information stored in those WMEs (e.g. L2 represents the list of entities and their attributes that the player has knowledge about) remain the same (e.g. L4 stores the exact same information as L2). The state of the world, NPC's, and the player are all copied over into the modeling state for prediction.

4.1. Model Design

The director has an internal probabilistic rule-based model of the player's behavior that is specified by the author (i.e. the model is domain-dependent and need be created by the author as a programmer) which is shown in Figure 3. This model is intended to represent a general hypothesis of how a player would behave in the story world. Player behavior in the *Haunt 2* environment is strictly comprised of physical actions, which makes modeling that behavior a more approachable problem as opposed to, for example, dealing with the player engaging in verbal conversations and attempting to predict speech acts. This approach may be expanded to deal with verbal interactions (e.g. discourse plan recognition), but is not within the current scope of IDA. The model used for *Haunt 2* is composed of both an operator hierarchy very similar to the hierarchy used to define the HauntBot agents and the hypothesized knowledge base of the player. This hypothesized knowledge base is the main source of information that the director has

about the individual currently playing the game; the player model, as currently designed, is a general model of player behavior. The operators used in the *Haunt 2* player are a subset of those used by the *HauntBots* for autonomous behavior in the game. Operators do not decompose into atomic actions (e.g. move forward one step), but instead only decompose into executable abstract actions, such as *move-to-area*. The player is modeled to instantaneously move to the target area with some associated cost in time when *move-to-area* is selected and applied. This is a rudimentary player model used to show the benefits of a predictive director. Future work entails research on defining more rigorous and adaptive models.

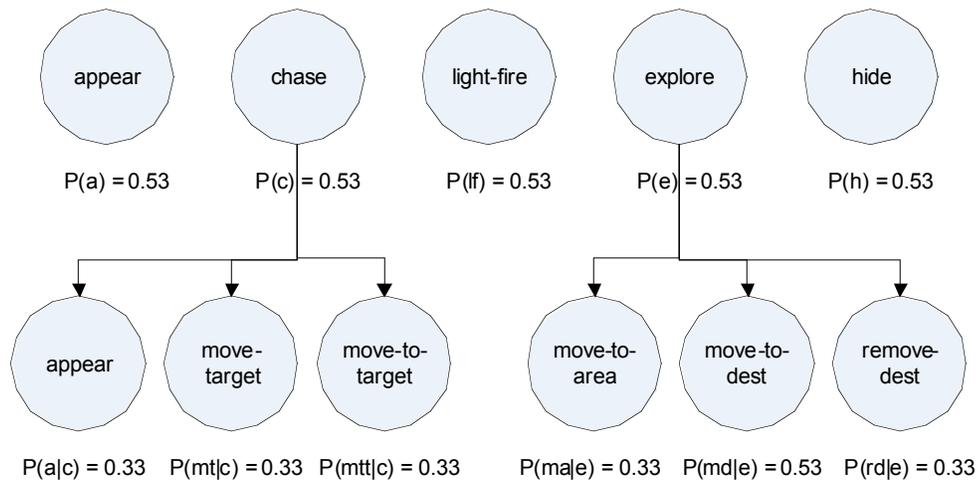


Figure 3. Probabilistic model of player behavior in *Haunt 2*.

Model operators are run on the simulated environment just as if the director agent were proposing, selecting and applying operators in the real world (as the player). When choosing which model operator should be selected at a given time, the director chooses one of the operators that are applicable in the current situation based on probabilistic control knowledge per the Soar decision cycle described in Section 3.3. For example, when all top-level operators in the model subset shown in Figure 3 are applicable, *explore* will be selected 60% of the time. The operators that have conditions matching current WMEs are proposed. This provides a more

functional model that represents a probabilistic space of players rather than a more rigid deterministic model. This also allows the director to estimate the probability of the occurrence of model actions.

The director runs this player model on the simulated world, executing rules to simulate how the world would respond to the player's actions and observing what plot elements are affected by the model's actions. The model returns a tuple $M \rightarrow (R, P)$. R is the modeling result, which is either a "success," meaning that an active plot point's preconditions are fulfilled by player behavior, or a "failure," indicating that no active plot points are fulfilled. P is the probability of the sequence of operators selected in that modeling run, $P(\text{Run}_x) = P(\text{Operator}_{x1}, \text{Operator}_{x2}, \dots, \text{Operator}_{xn})$. For example, after the game has begun and the player has executed a few actions, the director creates a copy of the world, runs the player model on that copy, and returns the result that the player will remain in that room (probably examining objects) until the next plot point's timing constraint is violated in the simulation.

Once a modeling run is completed, the director goes through the entire process again, executing a Monte Carlo simulation until some author-determined limit ρ is reached. Once the simulation is completed, the director computes the probability of the player fulfilling plot content, $P(F)$, which is defined as:

$$P(F) = (s/n_s) - (f/n_f)$$

Equation 1.

where s is the sum of P values across runs with $R = \text{success}$, n_s is the number of successes, f is the sum of P values across runs with $R = \text{failure}$, n_f is the number of failures, and $-1 \leq P(F) \leq 1$. Therefore, the value $|P(F)|$ equals the amount of confidence we have in R occurring in the story world, according to the player model.

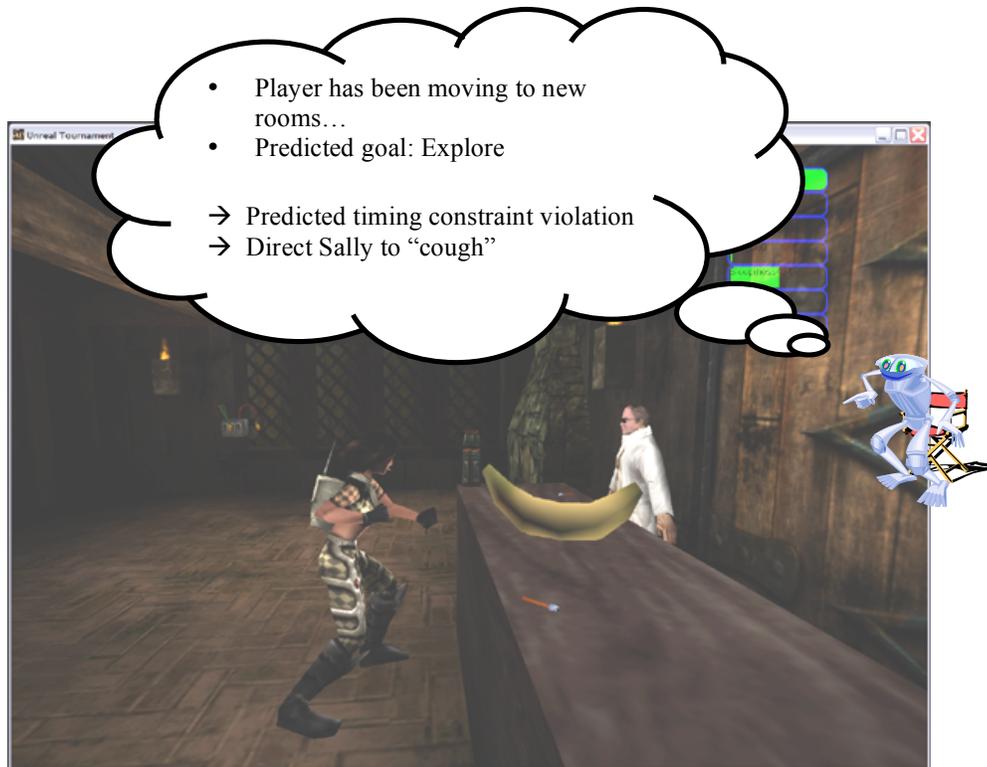


Figure 4. An example of predictive modeling being used.

The result of modeling, $P(F)$, is used to determine if the director should preemptively direct. If $P(F)$ is above some author-defined threshold α , which represents the author’s desire for how frequent the director should direct in a particular domain, then the modeling is considered to indicate that the player is likely to contribute to future plot content. If $P(F) < \alpha$, then the director will decide to preemptively direct the world because the story’s progression is likely to be hindered.

In Figure 4, we can see that the player, Innkeeper, and Sally should all be near each other. The player knows where the dead body is and should be invisible. Once the preconditions of this plot point are fulfilled, the characters are to reveal more information about their relationship to the player. Suppose that, at this point in the experience, the plot point reflected in Figure 4 has just become active. The player was staying near the other characters for awhile and has now

begun searching for clues through the different rooms in the building after the last plot point has ended. The director queries the predictive model, which hypothesizes with a high degree of certainty that the player is exploring and will continue to explore for some time, not entering the lounge as required by the plot point content. The director concludes that there is a high probability of a timing violation occurring and should therefore execute some direction in the environment. Since this timing violation has yet to occur, the director selects an action that is high in subtlety and a little low in terms of effectiveness, such as directing one of the characters to cough loudly. The director executes this command and the player, who is exploring the world, may be attracted to this new input and go to the room where the other characters are. Without the use of prediction, the director would have had to either wait until a timing violation had actually occurred or execute a subtle action at every possible opportunity.

4.2. Connecting Modeling to Director Action

The section above shows how predictive modeling is used by the director to decide whether or not it should execute direction. The director's next step after deciding *to direct* is to decide which direction is most appropriate (i.e. *how it directs*) based on the modeling results. Director actions are rated by the author in terms of several dimensions, the most important (and currently only ones used by the architecture) being *subtlety* and *effectiveness*. The dimensions that the author uses can be specified on a scale of 0 to 1. The selection of dimensions and rating of actions is up to the author (e.g. she may decide to rate actions based on how many characters are involved), which is similar to author-rating in MOE (Weyhrauch 1997). This role for the author allows her to encode her artistic ideal of what is important for selecting director actions. I have decided on *subtlety* and *effectiveness* to help illustrate the benefits of using predictive modeling

in an interactive drama. In addition to these ratings, different scoring functions are defined to reflect the author's bias in how the director should respond to the modeling result.

Scoring functions for IDA are defined as:

$$\text{score} = (S_1 * s_1 + S_2 * s_2 .. + S_n * s_n) / n$$

Equation 2.

where n is the number of dimensions used, s_1 to s_n are the ratings used by the author, and S_1 to S_n are the weights used to reflect the author's desire for the particular scoring function. These weights are assigned at run-time to reflect the appropriate weightings for a given situation. The specific scoring function used for *Haunt 2* is:

$$\text{score} = (S * \text{sub} + E * \text{eff}) / 2$$

Equation 3.

where n is the number of dimensions used ($n=2$ in this case), *sub* and *eff* are the subtlety and effectiveness ratings, and S and E are weights. If $S > E$, then subtlety is more preferred than effectiveness for that function. For example, if $\alpha < P(F) < \alpha + \Delta_{\text{pos}}$, where α is an author-defined threshold that defines the line between a positive and negative result and Δ is some small value used to indicate that the result was only a failure or success by a small fraction, then the author might write a scoring function to highly prefer subtlety over effectiveness when the director is heuristically choosing a director action. If $\alpha - \Delta_{\text{neg}} < P(F) < \alpha$, then subtlety might be a little less preferred than effectiveness. If $P(F) < \alpha - \Delta_{\text{neg}}$, then effectiveness may be preferred heavily over subtlety. These relations are designed to encode the relationship between modeling result and director action selection, but are not based on any preexisting theory.

The important factor in this design is that the score for a director action is dependent on the current situation. If a situation is urgent (i.e. requiring reactive direction), then the weights will be assigned to favor effectiveness, thus affecting which direction has the highest score. If the

predicted problem occurs off in the future, then the weighting will naturally shift towards more subtle measures. This design allows a connection between the urgency of a boundary problem and the direction that is selected to address it.

The *cough* direction selected in Figure 4 is an example of the selection of director actions based on score. If modeling returns a success ($\alpha < P(F)$), but the confidence in that result is low ($P(F) < \alpha + \Delta_{\text{pos}}$), then the weights are assigned as $S = 0.8$ and $E = 0.2$ to reflect the preference for more subtle measures. If modeling returns a failure ($P(F) < \alpha$) with high confidence ($P(F) < \alpha - \Delta_{\text{neg}}$), then the weights are opposite: $S = 0.2$ and $E = 0.8$. If the possible actions that are shown in Table 1 are the ones proposed by the director, then *cough* would be selected if $S = 0.8$ and $E = 0.2$. If reactive direction were being executed, then the more effective strategy, *actor-to-area*, would be chosen out of this set of possibilities.

proposed direction	subtlety	Effectiveness
cough	0.7	0.5
actor-to-area	0.3	0.8
create-sound	0.8	0.1

Table 1. Proposed director actions for attracting player in Figure 18.

The combination of reactive and preemptive direction provides IDA with a unique approach to addressing the boundary problem. While this method does make use of a specific predictive model to show prediction's value in IDA, it uses a simple hand-authored model of general player behavior. More adaptive or empirically-derived models would most likely better demonstrate IDA's benefits; however, if experiments can show that this basic approach to predictive

modeling is useful, then more accurate models must be at least as useful, if not much more so. The following section evaluates the use of this model.

5. Evaluation

One of the contributions of this work is the quantitative evaluation of the architecture's components' usefulness, specifically the added value of using a director that employs preemptive direction to avoid boundary problems. Few examples exist to date of experimental evaluations of the approaches used in interactive dramas (Nelson and Mateas 2005; Weyhrauch 1997). MOE's evaluation consisted of simulating types of players playing in a simulated game world; no actual game was actually created. Nelson and Mateas later implemented MOE's approach in an actual game system and found conflicting results as to the effectiveness of using the adversarial search methods used in MOE.

Very little work has been done on evaluating interactive drama systems with actual player inputs. I have created an experimental design that takes one step towards validating the approaches used in IDA with real player testing. A significant constraint to this design, however, is cost. While testing *Haunt 2* with a large set of human players would be ideal, the constraints of this project have led us to take a more economical approach. My goal in evaluation is to conduct a low-cost, quantitative experiment that will support my claim of preemptive direction being a useful alternative to reactive direction in an interactive drama by testing the effectiveness of my implementation of IDA. The experimental design used for this evaluation is not the most rigorous possible but is a low-cost step in a quantitative evaluation of preemptive direction in an interactive drama. As opposed to testing many different types of players with the system, I have investigated the use of *player archetypes* to define different behaviors for a single playtester.

The design involves the defining of player archetypes, which are used to define how the experimenter should behave while playing the game. The experimenter then plays through the game, taking on the different game personae defined in the archetypes. The data recorded from playing as these different archetypes, which are defined in Section 5.1, with preemptive direction on and off are then used in the evaluation. The comparison between these two groups is done by comparing the mean number of timing violations that occur, the number of preemptive and reactive directions that are executed by the director, and the average subtlety of those director actions.

5.1. Defining Player Archetypes

Player archetypes are canonical definitions of player behavior that are used to represent players across the spectrum of likely player types. Some work has been done in identifying player archetypes for Multi-User Dungeons (MUDs) which identifies four extreme player archetypes: *Achiever*, *Explorer*, *Socializer*, and *Killer* (Bartle 1996). The use of archetypes in this evaluation is meant to explore a subset of the space of possible player types. Bartle uses archetypes to define a player space along the dimensions of *action* and *focus* (e.g. whether or not the players focus more on acting on players vs. the world). This space describes how players behave in the world and what their behavior is focused on. Each archetype is intended to represent an extreme in that space, with most player behaviors existing at points somewhere in between. Archetypes are used for evaluating IDA's effectiveness as a means of representing the edges of the space of typical player behavior in *Haunt 2*. The steps for doing this are first to hypothesize what players may want to do in the world, then to attempt to map those to archetypal definitions.

In order to identify what archetypes could be used to define player behaviors in *Haunt 2*, a list of possible player goals was hypothesized along with a mapping from each goal to the game actions involved in achieving those goals: *Explore* (move to new rooms or rooms with new attributes), *Chase* (follow NPCs around while visible), *Learn* (listen to NPC conversations), and *Coax* (coax NPC to a room).

These goals and actions are used to describe a set of proposed player archetypes. The set of archetypes below represents the plausible behaviors that are defined by the set of goals above:

- GENERAL (a behavior that reflects that user model's preferences)
- EXPLORER (focuses on exploring the physical space)
- PLAYER (focused on figuring out the game)
- TESTER (pushes the limits of the system, scare the NPC's a lot, etc.)
- SOCIALIZER (hangs around the characters and tries to listen to them / affect them)
- SCARER (scares players but doesn't really do much else)

These archetypes can further be defined by the goals and actions listed above, providing us with a mapping from archetypes to actual agent behaviors:

	Explorer	Player	Tester	Socializer	Scarer	General
Explore	High	High	High	Medium	Low	Medium
Move to new inputs	High	High	Medium	High	Medium	Medium
Chase	Low	Low	High	Low	High	Medium
Learn	Medium	High	Low	High	Medium	Medium
Coax	Low	Medium	Medium	Medium	Low	Medium

Table 2. Preferences defining player archetypes.

5.2. Experimental Design

A subset of these archetypes was selected for experimentation: *Explorer*, *Scarer*, and *General*. These archetypes do not represent the complete space of possible player behaviors, but do sufficiently cover wholly different parts of the space defined by the archetypes above based

on the different emphases on possible domain-specific player goals. This design defines one experimental group, *modeling*, and one control group, *no-modeling*, to test the usefulness of preemptive direction. The *modeling* group involves the director operating with all of its capabilities intact; it performs story, reactive, and preemptive direction. The *no-modeling* group has all capabilities intact except for preemptive direction. The director does go through the process of modeling but does nothing with the results of the prediction, so there is no obvious time advantage for this group by not spending decision cycles on modeling.

The variables involved in each run are: the type of archetype used (*arch*), the number of timing violations (*num_tv*), the number of directions executed by the director (*num_dir*), and the average & median subtlety of directions (*avg_sub* & *med_sub*). *Num_tv* indicates the number of times the player's behavior has caused a boundary problem. *Num_dir* gives a rough measure of how often the director has to interfere with the world based on player behavior - the less interference, the better. My hypothesis is that *num_tv* will be higher on average for the *no-modeling* group, since preemptive direction should help avoid at least some boundary problems. I also hypothesize that *avg_sub* should also be higher in *no-modeling* than for *modeling*, since modeling attempts to make use of more subtle direction. The director parameters were assigned as follows: $\alpha = 0$ (on a scale between -1 to 1), $\Delta_{\text{neg}} = -0.12$, and $\Delta_{\text{pos}} = 0.12$. These parameter assignments were tweaked during the design process to elicit the desired director behavior.

Experimental runs were done 15 times for each archetype in each group. A single run involved me, the experimenter, following the behavioral rules described for the selected archetype. The director logged each measured event (e.g. timing constraints) into a file. Each file was parsed with a simple parser program and the statistics for each experimental run were computed.

A more rigorous design would have been to encode these behaviors in an autonomous agent or better yet have another person or people conduct the runs instead of me. The agent approach was ruled out since the same bias would likely exist if I built the agent versus doing the runs myself. The second was ruled out for pragmatic reasons, since the time taken to conduct the runs was over forty hour's time. This design acknowledges the relaxed approach used in this part of the experiment and should be used to indicate where future work can improve on it.

5.3. Results

The key hypothesis to test was that *num_tv* would be significantly higher for the *no-modeling* group than the *modeling* group ($H_0: \mu_{\text{model}} = \mu_{\text{no_model}}$, $H_A: \mu_{\text{model}} < \mu_{\text{no_model}}$). The overall statistics of the 90 experimental runs can be seen in Table 3. A univariate analysis of variance was run on the collected data, comparing the means across variables for the *modeling* and *no-modeling* groups. The results of the comparison are displayed in Table 4. A significant difference was found between the means of *num_tv* in the *modeling* and *no-modeling* groups ($F=8.444$, $N=45$, $p < 0.01$). This finding confirms my hypothesis that the use of predictive modeling does have a significant difference on the number of boundary problems that occur. There is also an insignificant difference for *num_dir* ($F = 0.49$) and *avg_sub* ($F = .818$).

These results suggest that predictive modeling does have an overall positive effect by decreasing the number of timing violations experienced by the player. This is likely due to the use of directions that guide the player away from boundary problems with at least some moderate success. While there is bias in the experimental design due to me playing as the archetypes versus other people playing it, this is definitely a strong result ($p < 0.01$) that highlights the benefits of the approach.

Type		# of timing violations	# of directions	avg subtlety of direction
no_modeling	Mean	4.80	10.49	.358
	N	45	45	45
	Std. Deviation	1.73	4.62	.070
modeling	Median	5.00	9.00	.350
	Mean	3.67	10.69	.372
	N	45	45	45
	Std. Deviation	1.97	3.88	.0699
	Median	3.00	10.00	.371

Table 3. Basic statistics from experimental runs.

		Sum of Squares	df	Mean Square	F	Sig.
# of timing violations * Type	Between Groups (Combined)	28.900	1	28.900	8.444	0.005
	Within Groups	301.200	88	3.423		
	Total	330.100	89			
# of directions * Type	Between Groups (Combined)	0.900	1	0.900	0.049	0.824
	Within Groups	1,600.889	88	18.192		
	Total	1,601.789	89			
avg subtlety of direction * Type	Between Groups (Combined)	0.004	1	0.004	0.818	0.368
	Within Groups	0.428	88	0.005		
	Total	0.432	89			

Table 4. ANOVA table comparing means of *modeling* and *no-modeling*.

ANOVA tests were calculated within each archetype group to investigate what factors contributed to this positive result. The *general* and *chase* groups yielded no significant difference between the means ($F= 0.607$ and 0.455 respectively). The *explore* group's comparison, however, was overwhelmingly significant ($F= 25.27$, $N = 30$, $p < 0.01$). To confirm the *explore* group's affect on the results, a final analysis was done on the *explore* and *general* groups together. The difference in means for this test was insignificant as well. This result points to the *explore* group as being the main contributor to the overall ANOVA results.

This finding does bring into question the strength of the evaluation's results, but it is more an affect of the predictive model's ability to cover the different archetype behaviors rather than of the effectiveness of the preemptive approach. Ideally, a significant difference in means would be found in all archetypes. The lack of results found in *general* and *chase* are likely due to the predictive model's lack of coverage for these behaviors. The *explore* archetype is arguably the most similar in behavior to the predictive model; therefore, the model was more likely to be accurate for this archetype than the others in retrospect.

There were inconclusive results for significant differences in the number of directions ($F = 0.049$) and average subtlety of the directions ($F = 0.818$) across groups. My hypothesis is that *num_dir* in both groups should be roughly equal, since the director is preemptively directing for boundary problems that would otherwise occur and thus be directed reactively. The lack of a difference in *avg_sub* is a little more problematic. One possible reason for this negative result could be the fine grain that was used to rate the directions (i.e. all directions were rated between 0 and 1). Another more likely reason could be the lack of variety in rating the directions; I as an author did not use a wide spectrum of values for assigning *subtlety* and *effectiveness* ratings. In retrospect, paying closer attention to the ratings given to directions would have affected this result. However, the most important result by far is the significant difference in *num_tv* between groups, which was a positive one. This work does operate under the assumption that fewer boundary problems are better for the player; however, future work should empirically verify this assumption.

6. Discussion

Given the low-cost nature of the experimental design, there are several dependencies that may influence the results of this study. These dependencies reflect possible biases in the

different components of the IDA architecture: director bias; player bias; and story bias (which includes the story world, content, and synthetic character behavior). These components represent the major factors within the closed system of the experiment.

6.1. *Director Bias*

The director's behavior is dictated by the knowledge it gathers about the environment and the player, the parameter settings for modeling and using the modeling result, and the director strategies that are authored for the story domain.

6.2. *Knowledge Model*

The gathering of knowledge is a necessary function of the director that is unlikely to introduce bias into the experimental results, unless the knowledge model is authored incorrectly. The hypothesis of player knowledge could contain error (e.g. just because a player walks into the lounge does not mean that the player sees the match and knows that putting it into the fireplace will start a fire), but this error would be consistent across experimental groups. However, the predictive model does take the knowledge model as an input. Therefore, the accuracy of the knowledge model may affect the accuracy of the predictive model, which in turn may affect the decisions of when to preemptively direct and how.

6.3. *Predictive Model*

Just as the accuracy of the knowledge model may introduce bias, the accuracy of the predictive model also has a similar effect. The less accurate the model, the less reliable the director's decision is to preemptively direct or not, which in turn decreases the accuracy of the

comparison of timing violations between groups. IDA's model for player behavior in *Haunt 2* is discussed in Section 5.3.

6.4. Parameter Values

The results of IDA's evaluation depend on the author-assigned parameter values of α , Δ_{neg} and Δ_{pos} , which dictate director behavior. If the modeling result is below the author-defined threshold $\alpha + \Delta_{\text{pos}}$, then the director will perform preemptive direction. This threshold has an effect on the director's decision making process in terms of when to direct. If this threshold is very high (i.e. close to 1) the director will almost always decide to direct, since $P(F)$ is very unlikely to be consistently that high, if at all. Conversely, if the threshold is very low (i.e. close to 0), the director will seldom decide that the preemptive direction is needed.

In the case of a high threshold, the director behavior for *modeling* versus *no_modeling* would certainly differ in the number of directions executed. Unless the predictive model is extremely accurate, the *modeling* condition would essentially execute preemptive direction every time the model was queried, regardless of player behavior. The director would nearly always come to the conclusion that a boundary problem was going to occur since the modeling result would almost always be below the threshold for assuming that the story will continue. The *no_modeling* case would be unaffected, since this threshold only affects the decision to preemptively direct.

While $\alpha + \Delta_{\text{pos}}$ determines the threshold between directing or not, the sizes of $|\alpha - \Delta_{\text{pos}}|$ and $|\alpha + \Delta_{\text{neg}}|$ affect the average subtlety of the director actions. As Δ_{pos} increases, the average subtlety *and* number of directions should increase since the threshold $\alpha + \Delta_{\text{pos}}$ is larger and the number of uncertain modeling successes ($\alpha < P(F) < \alpha + \Delta_{\text{pos}}$) increases. Both *num_dir* and *avg_sub* in the results may have been affected by the assignment of these values.

As stated earlier, these values were tweaked during the design phase to have the director's behavior match the author's desire (i.e. my desire) of what the director should do and when. This suggests that the parameter values may be dependent on the domain, the specific author, the plot content, and / or the director actions available.

6.5. Director Strategies

The director strategies authored for a story domain determine how the director can attempt to affect player behavior. The effectiveness of reactive or preemptive direction could be weakened if the authored strategies do not adequately cover the space of rated dimensions (e.g. subtlety and effectiveness). For example, if the director actions are all very subtle and not effective, then the story may not even continue when reactive direction is executed since the system has no effective strategies to put the story back on track. Conversely, if only effective strategies are used, then timing violations would still be addressed, but the average subtlety of direction would decrease, therefore reducing the believability of the experience. As shown in Appendix A1, there are IDA director actions that are spread across the two rating dimensions for *Haunt 2*, which avoids the extreme conditions described here.

6.6. Player Bias

The same experimenter, computer (a Dell Inspiron 8600 laptop), and experiment location was used for all tests as a control. Any possible bias that was introduced by using the experimenter as the subject is the result of the archetype's sensitivity to direction, the experimenter's inherent sensitivity to direction, and the experimenter's awareness of the current experimental group used in a given run.

6.7. *Archetype and Individual Sensitivity*

The weight coefficients S and E represent the author's prediction of the subtleness and effectiveness each direction has on the player. These coefficients directly contribute to which director action is selected in a given situation. The assigned values for these coefficients are unlikely to be accurate across all player archetypes. For instance, creating a new sound in a nearby room might be highly effective for the *explorer* archetypal player, but not nearly so for the *scarer*, who is not as focused on exploring new information. As the experiment was executed, this effect was approximated by the experiment, but did not necessarily reflect the true values of the coefficients. Ideally these values would be determined from player testing and adjusted depending on the player's apparent playing style.

This discussion is part of a larger issue of players' general susceptibility to direction. Different players of the same behavior type (e.g. *explorer*) may have different responses to specific director actions. The factors that may influence susceptibility include game experience, playing style, and gender (Heeter and Winn 2005). For example, experienced computer gamers may easily notice when the system is trying to avoid or deal with boundary problems, thus decreasing the believability of the experience and perhaps the effectiveness of using direction, while novices of the same archetype are less aware of the system's manipulations.

These individual differences within archetypes were not modeled in the archetype behaviors definitions for this study. While acting as the experimenter in this study, I attempted to respond to directions as an archetype would. However, my responses to direction remained constant while playing as that archetype; individual differences were not modeled. This lack of variety is most likely different from reality. Therefore, the data collected in this experiment is an idealized

data set; there is no variability between runs of how likely a player within an archetype will respond to direction.

6.8. Awareness of Experimental Group

If the subject knew whether preemptive direction was on or off for a particular run, the subject's behavior could be biased. Therefore, I was only aware of the desired archetype behavior during any given run. Whether or not preemptive direction was turned on was not directly known by me, nor did I make a conscious effort to surmise this. The only possible bias here was my unconscious determining of the experimental group and letting that affect my behavior. The only means of removing that bias will be to have a more formal experiment with human subjects not involved in the creation of the system.

6.9. Story Bias

Authored content (e.g. plot points, predictive model and synthetic character behaviors) can bias the experience to be more amenable to certain player archetypes versus others. If the content matches particularly well with a specific archetype's goals, then that archetype would be expected to have, on average, fewer timing violations than the others. For example, if the *Haunt 2* story was focused on a tale of the player chasing occupants out of the building, then the *scarer* archetype will be less likely to commit timing violations since the archetype's typical behavior would be closer to the story goals. The archetypes were defined to cover the space of possible archetypal behavior for *Haunt 2* in an attempt to avoid this problem.

7. Work Cited

(1980). "Zork I: The Great Underground Empire." Infocom.

- (1983). "Planetfall." Infocom.
- (1984). "The Hitchhiker's Guide to the Galaxy." Infocom.
- (1993). "Myst." Brøderbund.
- (1998). "Grim Fandango." LucasArts.
- (1999). "Gabriel Knight 3: Blood of the Sacred, Blood of the Damned." Sierra Entertainment.
- (2002). "Syberia." MC2-Microïds.
- (2005). "The Indigo Prophecy." Atari.
- Adams, E. (2005). "You Must Play Façade, Now!" Gamasutra.
- Assanie, M. (2002) "Directable Synthetic Characters." *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*, Palo Alto, CA.
- Bartle, R. (1996). "Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs."
- Beal, C. R., Beck, J., Westbrook, D., Atkin, M., and Cohen, P. (2002) "Intelligent Modeling of the User in an Interactive Environment." *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*, Palo Alto, CA.
- Blumberg, B. and Galyean, T. (1995) "Multi-level Direction of Autonomous Characters for Real-Time Virtual Environments." *Siggraph*.
- Blumberg, B. and Galyean, T. (1997). "Multi-level Control for Animated Autonomous Agents: Do the Right Thing...Oh, Not that." *Creating Personalities for Synthetic Actors*, R. a. P. Trappl, P., ed., Springer.
- Bringsjord, S. and Ferrucci, D. A. (2000). *Artificial Intelligence and Literary Creativity*, Lawrence Erlbaum Associates, Mahwah, NJ.
- Cavazza, M., and Charles, F. (2005) "Dialogue Generation in Character-based Interactive Storytelling." *1st Artificial Intelligence and Interactive Drama and Believable Agents*, Marina del Rey, CA.
- Cavazza, M., Charles, F., and Mead, S. (2002) "Sex, Lies, and Video Games: an Interactive Storytelling Prototype." *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*, Palo Alto, CA.
- Fahlstein, N. (2005). "Understanding Fun - The Theory of Natural Funativity." *Introduction to Game Development*, S. Rabin, ed., Charles River Media, Inc., Hingham, MA.
- Fairclough, C. (2004). "Story Games and the OPIATE System," University of Dublin – Trinity College, Dublin.

- Field, S. (1994). *Screenplay: The Foundations of Screenwriting*, Dell Publishing, New York.
- Gordon, A. and Iuppa, N. (2003) "Experience Management Using Storyline Adaptation Strategies." *Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt, DE.
- Gratch, J., Rickel, J., André, E., Badler, N., Cassell, J., and Petajan, E. (2002). "Creating Interactive Virtual Humans: Some Assembly Required." *IEEE Intelligent Systems*, 54-63.
- Heeter, C. and Winn, B. (2005) "Gender, Playstyle, and Learning: Constructing In-Game Measures of Playstyle." *FuturePlay*, East Lansing, MI.
- Kelso, M. T., Weyhrauch, P., and Bates, J. (1993). "Dramatic Presence." *Presence: Journal of Teleoperators and Virtual Environments*, 2(1), 1-15.
- Klesen, M., Kipp, M., Gebhard, P., and Rist, T. (2003). "Staging Exhibitions: Methods and Tools for Modeling Narrative Structure to Produce Interactive Performances with Virtual Actors." *Virtual Reality*, 7(1), 17-29.
- Laird, J. E. (2001) "It Knows What You're Going To Do: Adding Anticipation to the Quakebot." *5th International Conference on Autonomous Agents*, Montreal, CA, 385-392.
- Laurel, B. (1986). "Towards the Design of a Computer-Based Interactive Fantasy System," Ohio State University, Columbus, OH.
- Loyall, A. B. (1997). "Believable Agents: Building Interactive Personalities," Carnegie Mellon University, Pittsburgh, PA.
- Magerko, B. and Laird, J. E. (2004) "Mediating the Tension Between Plot and Interaction." *AAAI Workshop Series: Challenges in Game Artificial Intelligence*, San Jose, CA.
- Magerko, B., Laird, J. E., Assanie, M., Kerfoot, A., and Stokes, D. (2004) "AI Characters and Directors for Interactive Computer Games." *16th Innovative Applications of Artificial Intelligence Conference*, San Jose, California, 877-883.
- Mateas, M. and Stern, A. (2000) "Towards Integrating Plot and Character for Interactive Drama." *Fall AAAI Symposium: Socially Intelligent Agents: The Human in the Loop*, North Falmouth, Massachusetts.
- Mateas, M. and Stern, A. (2002) "A Behavior Language for Story-Based Believable Agents." *AAAI Spring Symposium Series: Artificial Intelligence and Interactive Entertainment*, Palo Alto, CA.
- Mateas, M. and Stern, A. (2003) "Facade: An Experiment in Building a Fully-Realized Interactive Drama." *Game Developer's Conference*, San Francisco, CA.

- Meehan, J. (1981). "Overview of TALE-SPIN." Inside Computer Understanding, R. C. Schank and C. K. Riesbeck, eds., Lawrence Erlbaum Associates, Hillsdale, NJ.
- Nelson, M. and Mateas, M. (2005) "Search-based Drama Management in the Interactive Fiction *Anchorhead*." *1st Artificial Intelligence and Interactive Digital Entertainment Conference*, Marina del Rey, CA.
- Newell, A. (1990). *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts.
- Payne, A. (2004). "Sideways." London, M. and Parra, G.
- Reilly, W. S. (1996). "Believable Social and Emotional Agents," Carnegie Mellon University, Pittsburgh, PA.
- Riedl, M. and Stern, A. (2006) "Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training." *15th Conference on Behavior Representation in Modeling and Simulation*, Baltimore, MD.
- Sgorous, N. M. (1999). "Dynamic Generation, Management, and Resolution of Interactive Plots." *Artificial Intelligence*, 107(1), 29-62.
- Szilas, N., Marty, O., and Réty, J. (2003) "Authoring Highly Generative Interactive Drama." *2nd International Conference on Virtual Storytelling*, Toulouse, FR.
- Tambe, M. and Zhang, W. (1998) "Towards Flexible Teamwork in Persistent Teams." *Proceedings of 3rd International Conference on Multi-Agent Systems*, Paris, FR.
- Turner, S. R. (1994). *The Creative Process: A Computer Model of Storytelling and Creativity*, Lawrence Erlbaum Assoc., Hillsdale, NJ.
- Weyhrauch. (1997). "Guiding Interactive Drama," Carnegie Mellon University, Pittsburgh.
- Young, R. M., Riedl, M., Branly, M., Jhala, A., Martin, R. J., and Saretto, C. J. (2004). "An Architecture for Integrating Plan-based Behavior Generation with Interactive Game Environments." *Journal of Game Development*, 1(1), 51-70.
- Zubek, R. (2005). "Hierarchical Parallel Markov Models for Interactive Social Agents," Northwestern University, Evanston, IL.