Building an Interactive Drama Architecture

Brian Magerko and John Laird

University of Michigan
Advanced Technology Laboratory
1101 Beal Ave., Ann Arbor, MI 48109
magerko,laird@umich.edu
www.soargames.org

Abstract. Previous approaches to interactive drama with a pre-specified plot structure have relied on limiting the User's interactions with the world. The architecture proposed in this paper, IDA, attempts to create an experience driven by a pre-authored plot, without explicitly limiting the User's actions, some of which may work directly against the intent of the plot. In order to allow the User as much freedom as possible, while still maintaining the narrative, we introduce an omniscient agent, the story Director, which performs three main tasks in order to protect the narrative's integrity: plot verification and behavior monitoring to make sure the user's behavior hasn't harmed the future narrative and is consistent with timing constraints of the plot, predictive modeling of the User to detect such errors before they occur, and preemptive direction to modify the behavior of the characters and the world to encourage the User to move along the narrative path and avoid complete plot failure.

1 Introduction

Building interactive storytelling systems has been a growing topic of research over the past decade. Works such as MOE [15] and FACADE [11] of the Oz project, DEFACTO [14], and the MIMESIS architecture have pushed the envelope of traditional narrative into a new medium: the automated interactive storyteller. These systems have focused on offering the User a compelling dramatic experience with the User as a main character in the narrative. Other approaches, such as Marc Cavazza's work [4], have explored user interaction within a storytelling system by moving the User to the role of an observer that can change the world as the story progresses, and thus changing the story's outcome. An example of this kind of interaction in the commercial computer game world is Will Wright's "The Sims," a game focused on the User affecting and managing intelligent agents' lives as they go about their daily routines. These approaches can be categorized along a spectrum of how much control the human author of the system is given over the story. Systems like FACADE and MIMESIS rely heavily on the specification of plot by a human. Further along the spectrum are projects like DEFACTO [14] and BRUTUS [3] that allow a human to logically specify dramatic principles and rules for how the world behaves, but lets the plot emerge from the User's interactions with the world. Our research focuses on providing the best of both worlds: developing a system that allows for specification of plot by human author while at the same time allowing user interactions to fill out plot content.

The experience our approach attempts to achieve is a drama presented with a *high degree of interactivity*, where the User's experience approximates "acting as they want to act" in the world, giving them as much freedom as possible in their interactions. This degree of user flexibility allows the User to execute actions or behaviors that may work *with* or *directly against* the plot description. Working against the plot means that the User may execute actions in the world that directly conflict with the pre-written story. Allowing these kind of actions is key to offering a sense of realism and more of a connection between the User and the experience he has via the character he controls in the drama.

There is an obvious tension between offering the Writer the amount of expression and control needed to fully communicate an artistic vision, and the presumed User desire for interactivity. His behavior cannot be known beforehand, like characters in a traditional narrative. Having no concrete representation for how this character will behave presents a dilemma for writing and presenting a coherent and interesting narrative. We are focusing on incorporating a predictive User model as an aid in keeping him within the drama desired by the Writer. By using a predictive model of User behavior, we can proactively avoid possible conflicts between User behavior and the plot in a realistic and believable fashion. As the User walks through the narrative, our system constantly asks such questions as "Is it possible for the plot to progress now?", "Is the User likely to reach the next story element written by the Writer?", and "If this is unlikely, how can we change the world to nudge the User back on track in the narrative?" This paper will discuss the needs of an interactive drama (or ID) with a high degree of interactivity, the approach we have taken to satisfy those needs, and finally an example of a working system using our approach.

2 Requirements for an Interactive Drama

Expressivity. A simple view of expression in narrative is that a particular narrative event can be broken down into describing WHAT happened, WHO was involved, WHEN the event happened, WHERE it took place, and HOW or WHY is occurred. Though there are possibly alternative ways of breaking down scenes or events in a narrative, this is certainly an acceptable one. Any representation that is used to describe a narrative in an ID should provide for the same degree of expressivity. A Writer should be able to construct a scene with an obvious means for communicating the aspects above, such as WHAT happens in the scene and WHEN.

Variability. A main facet of interactive drama as we define it is that the "interactive" aspects of a system provide for the User's actions and behavior to have a direct effect on plot content. For instance, if the User is supposed to meet a particular actor, the plot should be capable of specifying that, wherever the User decides to go in the world, he will wind up meeting that actor in a sensible fashion. By providing this type of connection between User behavior and plot content, the system offers variable experiences for the User, given the right plot construction. As the User makes different choices from experience to experience, the narratives in those experiences should have notably different content.

Flexibility. We want to provide the user with as much "open-ended interaction" as possible. This means constraining the User's experience only when necessary. The plot representation should provide for constraining the User's experience only in behavior that is relevant to the plot. In other words, the User should feel like he can behave how he wishes, and the plot will flow naturally from that behavior.

Variety in Degree of Autonomy. As Mateas and Stern [11] have pointed out, relying on strongly autonomous agents to perform a narrative leads to issues with coordination, plot representation, and overall performance of a story. While we agree that depending on strongly autonomous agents is not necessarily the best approach, we don't want to commit to a particular weaker level of autonomy either. In some plot situations, we can imagine wanting to specify actor behavior at a very fine level. In other situations, exactly what happens may not be relevant, so the Writer may give the actors much more freedom to choose their own behavior. Providing a representation that allows for varying degrees of specificity in actor behavior is a key requirement for our architecture.

Categorization of User Behavior. As we have noted above, the User should be capable of exhibiting behaviors that may work with the narrative, but also conflict with it. It is imperative that the system can clearly distinguish between the two. Once User error can be classified as "ideal" behavior or as an "error," the system can then behave appropriately. Systems such as MIMESIS [16] and the Oz Project [10] have directly addressed this problem of recognizing User error. The flip-side of the problem is, once User error has been recognized, what sort of reasoning needs to be done to have a narrative consistent with both this behavior and the plot as it has been written by the Writer. Dealing with User error is one of the main topics this paper will address.

3 An Interactive Drama Architecture: IDA

We have designed an architecture that provides for building interactive drama experiences, which we called an Interactive Drama Architecture, or IDA. Figure 1 illustrates the basic connectivity between the human element, namely the User and Writer, and the software components of the architecture. IDA relies on a central agent, the Director agent, which takes the initial state of the world and a plot description pre-authored by the Writer as initial inputs. The Director's role is to monitor both the User's behavior and plot, making sure that the narrative runs as smoothly as possible. We will go into more detail about each of the main components of the architecture below.

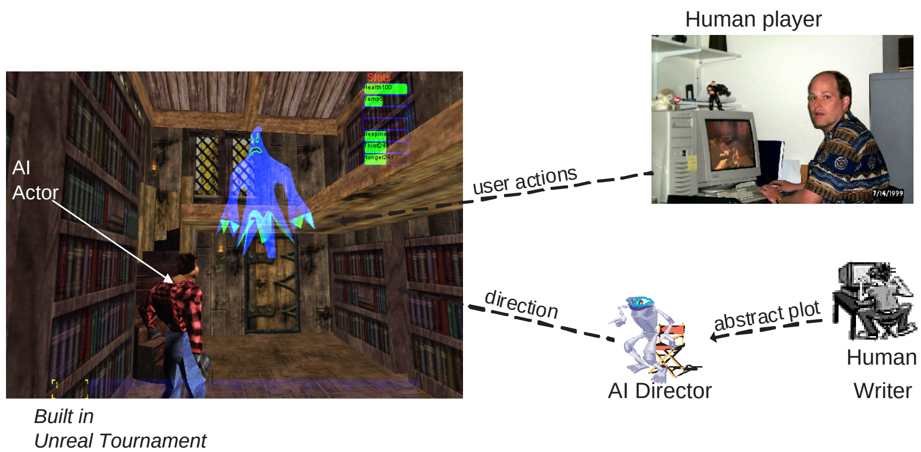

Fig. 1. The Interactive Drama Architecture, IDA.

3.1 The World

We have constructed a dramatic experience, called Haunt II [6], with the Unreal Tournament 3-D engine. This world offers the User a first-person view of the world, though that is not necessarily a vital of our architecture. The narrative takes place in a haunted house of sorts, where the User awakes to find himself as a ghost, trapped in this house with no idea how he got there or why. This world will be used in subsequent examples, but our architecture is by no means solely applicable to this domain. Our approach is intended to be far broader than only pertaining to the story we are using as our main example.

3.2 The Writer

When we describe the need for a human author, we are assuming that there is a narrator who wishes to communicate some artistic vision directly to the User. IDA includes a representation language, which will be addressed later in this paper, for the Writer to author a plot. Since the actual written plot is the only requirement for our approach, an actual human is not necessary. An intelligent, autonomous authoring system could be used to write the plot, which would then serve as the input to the system. While this is feasible, it is much more fruitful to consider human authors and how to deal with their plots before tackling the additional problem of automated authoring of plots for an ID.

3.3 The Director and Actors

IDA's Director agent is at the heart of maintaining the narrative as the User progresses though it. It takes the initial state of the world and authored plot as input. Once the experience begins, the Director has a dual role.

Its first responsibility is to direct the intelligent actors that populate the world. The characters are autonomous agents that have pre-defined characteristics to describe them physiologically, mentally, and in the future perhaps even emotionally. As the Director comes across plot content relevant to a particular actor, it has the job of giving that information to the actor. A different paradigm would be to simply give the entire plot to all of the actors, but it is difficult, if not impossible, to both maintain a global view of the world and make coherent decisions about plot direction in a distributed fashion.

The second role of the Director is to serve as a monitor of User behavior. As we have pointed out before, the User may execute actions that will work with or against the intent of the plot. We can categorize User behavior in one of three categories: *ideal*, *acceptable*, and *error*. When the User exhibits ideal behavior, he is contributing to the plot content in some meaningful manner and the Director needs to take note of it. That may simply be checking off that a User has entered a particular state, or may involve actual creation of plot content, which is discussed below. If the User's actions cannot be determined to be harmful or contributing to the plot, then the behavior is categorized as acceptable and nothing more should be done. If the User is committing an error, then it is the Director's job to recognize this as error and dynamically change the world and/or plot in order to incorporate this new action into the narrative. How the Director approaches this task is closely tied to the plot construction and will be discussed thoroughly in Sect. 4.

The third Director role is to instantiate content in the plot structure as the User moves throughout the world. Our plot representation allows for an incomplete specification of plot content; "holes" are left for the User's interactions to fill. The Director should take not when those holes are filled and instantiate them in the plot specification.

4 A State-based Model of Plot

While other projects have applied planning-based languages to narrative, such as FACADE, MIMESIS, and the story-generation systems TAIL-SPIN [12] and UNIVERSE [8], we have decided to approach story representation as a state-based model. In the state-based model, there is an inherent difference between representing the author's desired User experience (what the User *should do*) versus a prediction of the User's behavior (what the User *is likely to do*). Each scene in a plot is represented as a graph of desired states to be reached, as shown in Fig. 2. Representing plot as a graph structure has been used before, such as in the real-time selection of plot events [15] or the dynamic resolution of plot via social rules and dramatic principles [14]. The final goal states for a scene mark the end of the scene, just as the initial state describes how the scene should begin. If the Writer wishes to create a more complex and varied plot, he may describe multiple goal states to add a larger degree of variability in the narrative. A state is marked with a transition from one state to another, which provides a relative ordering of how states should flow temporally. Each transition is marked with a depth limit d, which provides timing information for how the plot events should flow. The scenes presented in this paper are provided, for simplicity, as a linear ordering of events to construct a single scene. However, in general we are providing for a fixed, linear connection of scenes as our representation. We are not focusing on the dynamic ordering of scenes, as has been done in the Oz project [15].

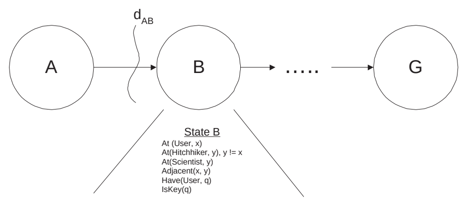

Fig. 2. Plot as State Transitions: a graph of partially-ordered plot events.

The contents of each state will be a connection of logical clauses and/or constraints, describing how the world should appear to the User. A plot state actually represents multiple possible states in the world; it is not a specific snapshot that describes everything in the world. Figure 3 illustrates the plot structure, as well as the search and prediction models at a particular transition that the Director will use. This example plot describes a world state in which the User is in a different room than the Scientist and Hitchhiker characters, who are in an adjacent room together. The User also has the *key* object in his possession. Elements of the plot may be specifically bound, variabilized, and/or constrained to be the member of a specific set. Variables may be shared across states or even across scenes, allowing for significant plot elements to be bound consistently for the duration of the narrative.

A simple state-transition model is not a rich enough system on its own to describe a directable experience. With only states and transitions specified, it is unclear what actions the Actors will take during those transitions. There is also no metric of "failure" across time. For example, if the User is expected to quickly go between a set of closely related states, how do we specify such timing? We allow the Writer to provide for timing constraints on transitions between states by specifying some limit d_{AB} for each connection between two states A and B. Figure 3 shows limits on each transitions between states. This limit simply represents a generic timing metric; what an appropriate metric actually could be is an open question in our research. We will use "number of actions" as a simple but useful metric for now. If the User executes too many actions, or even is *projected to take too many actions* (see Sect. 5, then the Director should take note of this failure and try to modify the User's behavior accordingly.

This representation also provides for directly specifying Actor behavior during transitions, as seen in the bottom of Fig. 3. This provides us a way of avoiding

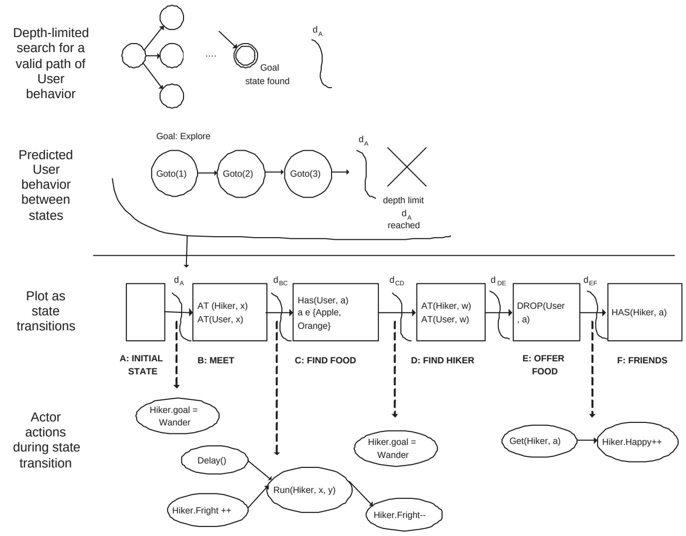

Fig. 3. An example scene: the User befriending the Hitchhiker by finding him food

problems that may come up when dealing with coordination between multiple actors, and implicitly helps us avoid coordinating with the User. For instance, suppose that between states the User has to meet another character in a room, receive a package from her, and then go elsewhere in order to enter the next plot state. Unless we have an intelligent manner of assuring that such an interaction integral to the plot will occur, the narrative may fail to be completed. We constrain the state representation to require any direct interactions with the User to be specified as state content. This approach has been used before in systems like FACADE, providing a more complicated framework of joint plans for specifying actor behavior.

5 Plot Verification and User Modeling

As we have mentioned earlier, the Director's role in plot monitoring is to *recognize* User error and to *act* on it. Now we will take the steps to define how we classify User error. "Error" can be seen as any behavior that does not fit within the context of the story. For instance, if our story specifies the User learning the contents of a particular book in the Scientist's laboratory, but that book is inadvertently destroyed as a consequence of the User's actions. These actions would be considered to be in error, as they have directly affected the story in such a negative manner as to make it impossible for the plot to continue as is. This error is directly plausible because of our desire for a pseudo-realistic environment in an interactive drama. Considering this kind of error, and how to act on it, in the virtual training community has also been addressed [9].

5.1 Search

We recognize User error by three different manners, treating the results of both methods as equivalent. The first is a search-based method. We ask the questions "Is the next state, or set of states, in the scene possible? are the scene goals still possible?" We answer this query via a depth-limited depth-first search, DFS, between states A and B, with the Writer-specified limit d_{AB} for the transition between A and B. The search space is constrained by the relatively small size of most scenes, keeping the time complexity of the search down to reasonable limits. We have explicit constraints on the plot specification that constrain this search to only the search through User actions, and not of all world actions. First, we have required any explicit interactions with the User to be specified through sequences of states. Second, any required or relevant Actor behavior is specified between states during their transitions. It is possible that the User may take some path between states A and B that was both unpredicted beforehand and requires interactions with actors. This is an issue we are currently considering to drive our work on using search methods for verification of the plot, or answering this question of whether or not a particular scene may still be experienced from the present to end.

5.2 User Modeling

Not only do we wish to ask if reaching the next state is possible, we should also be concerned with whether or not it is *likely* that the User will achieve this goal. This is a viable question because, while it is useful to know if the scene can possibly fulfilled, that does not necessarily mean the User is at all likely to take a successful path. If the User is likely to execute some behavior that is in conflict with scene content

We will employ some predictive user model M that will offer us a hypothesis of future User behavior. The model will return one of three types of a sequence of User actions: one that reaches the goal within d_{AB} , one that reaches d_{AB} without success, and one that reaches some terminal state, which may be treated equivalent to the second sequence type. The top of Fig. 3 shows an example of a prediction during the first state transition. The User model hypothesizes a user goal of exploration; he will attempt to move through several rooms, not reaching a goal state before executing less than and d_A actions.

If the predicted behavior reaches the next state from the current state within limit d_{AB}, then we may assume that a) the User is likely to achieve the next scene state and b) the User will do it within the temporal constraints of the experience, encoded by the limit d_{AB}. This means that the User is likely to follow the destiny laid out for him by the Writer. If the User veers from this prediction, or was predicted to not reach the next state, then the User is said to be in an *error state*.

The User model we have used thus far in our experiments is a relatively primitive one, constructed on our intuition. Its function is to serve as an example that predictive modeling can be used to the benefit of an ID system. We intend, however, to focus our future efforts on exploring what kinds of modeling are most effective. Predictive modeling has been sparsely applied to domains where narrative is involved [2]. We have identified at least two projects that have directly taken a step toward what we are trying to accomplish in IDA. Albrecht [1] applied Bayesian techniques for User goal-prediction in an online gaming system called Multi-User Domains (MUD's). Conditional probability tables were built up over time during data collection. These resulting Bayesian networks could be used at any time to predict what quest in the world the User was trying to accomplish. While this approach relied on there being explicit goals in the world, it certainly seems applicable to the domain of interactive drama and warrants exploration. The second project is the predictive model used in the University of Michigan's Quakebot [7], an intelligent opponent in the Quake computer game. The Quakebot observes user actions and performs an internal simulation of the current state of the world, asking itself "What would I do if I were the User?" As opposed to using itself as a model of the User, it could just as easily use any relevant Soar agent instead. Much like the use of model tracing in intelligent tutoring systems [5], we can see how make use of user model queries can lead us to change the content of a user-centered experience.

5.3 Behavior Monitoring

The third approach to recognizing User error is simply monitoring the User's behavior. If their behavior passes the limit d between states without reaching the state, then their behavior is classified as an error. If the User has reached this point, it means that the User model has failed to correctly predict an error. Something must then be done to encourage the timing constraints for the next scene to be met in the narrative.

6 IDA at Work

We have worked on building an IDA prototype in the Haunt environment. The synthetic characters populating the world are authored using Soar, a cognitive architecture that has been previously used for the authoring of intelligent agents in AI systems [13]. This architecture suits our needs for building the Director agent as we have described thus far, as well as nicely fits in with the Haunt infrastructure. Therefore, the Director agent in our prototype has been created as a rule-based Soar agent. It should be noted that this prototype, while definitely plausible, has not yet been finished. We fully expect to complete this stage in the near future.

Our prototype experiment intends to simply show that this architecture as created can not only offer a reasonable narrative for the User to take part in, but also that there may be some benefit to using a predictive user model. We have created a single actor in the world for the User to interact with, the Hitchhiker character. The User begins in the basement of the mansion, and the Hitchhiker outside the front door. The plot described in Fig. 3 is an initial input to the Director agent, along with the global state of the world (e.g. the states of the characters, including the User, items in the world, a map of the house, etc.). Each plot point is represented as a Soar operator. The operator is proposed if the conditions for the plot point are satisfied, such as AT(User,x) and AT(Hiker,x) in plot point B in Fig. 3. The application of the rule may include directions for performance, such as telling the Hiker that he should be frightened and run away. A non-linear ordering of plot points is easily achieved by describing several different operators that need not fire in any particular order.

Let's consider the first plot operator for the scene's initial state, which is represented in Soar as two productions. The first rule dictates that if there are two characters in the world, the User and the Hitchhiker, and they are both indoors, then the first plot point has been reached. The second rule describes what should happen after this plot point has been selected, namely instructing the Hitchhiker to have the Hitchhiker switch to the goal of exploring, if he has not done so already. Each plot point also has a declarative structure associated with it, describing what the timing limit for that point, actions the Director could take to encourage that point, and the corresponding ID label for that point.

The User's actions are received by the Director as messages from Unreal Tournament. Preprocessing on these actions may be done to generalize them, such as moving from one coordinate in a room x, to another room through a doorway y may be sent to the Director as MOVE-TO-ROOM(x,y). With each non-atomic action the Director observes, it will proceed to execute the search and user model as discussed in Sect. 5. Our prototype user model is a simple rule-based approximation of how the User would first behave in the world. It has goals such as EXPLORE, GET-ITEM, and GOTO-ROOM that describe what behaviors a User might have being first introduced to the world. Search control knowledge in the model takes form as operator preferences, specifying which operators are more preferred compared to others.

In order to perform the search / modeling in the Director, we have followed the example from earlier Soar programs that have done internal modeling of opponents in a game setting [7]. The Director's internal structure representing the world state is copied into a separate structure. The user model then is internally run on that simulated world, using the same rules it would execute as if it were operating an agent in Haunt itself. The model then runs until the Director either notices that a plot point may be reached or the plot point's limit is reached. The search process works quite similarly, except all of the preference knowledge is removed from the model, allowing for a depth-first search through possible User behavior.

Let's examine plot point C in Fig. 3 to understand how this Director agent works to elicit a coherent narrative using this plot structure. Suppose that during the User's explorations, he's not ventured into the area of the kitchen were the only food in the house is clearly visible. The Director will have a reasonably accurate model of the User's knowledge in this world, such as what he's done, who he has met, what objects he's aware of, etc. Before point C, the Director will conduct a search, indicating that it is possible for the User to go to the kitchen and get the fruit. It will then hypothesize about whether or not the User is likely to execute this behavior. Given that the model knows that the User does not know the food exists, nor that the Hitchhiker even desires the food, the user model will fail to predict the User will go to the food and get it within the allotted limit. The Director would then exit the modeling with a failure returned. A separate rule set exists to handle when a search or modeling effort fail. For now, they simply access the plot point's declarative structure and execute one of the specified Director actions associated with it. For instance, the Hitchhiker could be told to walk by the room the User is in, talking out loud about how hungry he is. This is a clear example of the Director reasoning about the User's behavior, the plot, and when the two may not overlap. What it is not capable of yet is more complex reasoning about what to do once a discrepancy is realized. In future generations of this agent, we would like to expand on both how the Director models the User and how the Director decides to change the world in response to its predictions.

7 Discussion

Putting all of these pieces to create a compelling narrative experience for the User is no easy task. There are many possible approaches not even touched on yet in the growing field of interactive drama. The purpose of IDA is to push forward the notion of building a more generalizable approach to building different narratives. This paper lays down the groundwork for such an architecture, although there are still several research issues that are open in the work. We believe that applying user modeling, as proposed by Beale at. el [2], is a novel approach for building interactive dramas that has a vast amount of promise. As the quality of prediction increases, the ability of the Director to guide the User's experience smoothly increases as well. As discussed earlier, how to generate an appropriate model for use in an interactive drama and how to best take advantage of that model shall be the focus of our research in the near future.

Aside from user modeling, we also have not thoroughly discussed how the system, once recognizing User error, guides the User's experience in an intelligent manner. The MIMESIS architecture avoids this problem by searching ahead for possible error and changing the world to avoid those errors at all costs. We believe in a more organic approach, both incorporating the User's actions into the story and attempting to influence him to follow certain paths. Our representation allows for a large degree of variability; large parts of plot content may be left uninstantiated at run-time. We have only begun to realize how the Director can take advantage of this fact to its potential. For instance, one can imagine telling a murder story and who the murderer turns out to be in the end is vastly affected by the choices the User has made along the way. Perhaps the Writer should be able to specify that the character who winds up being the User's closest friend is really the murderer, and the justifications for how he or she could have done it are dependent on what the User has witnessed, etc. While we are a far way from that degree of variability, we believe that IDA has taken a solid step toward it.

References

1. D. Albrecht, I. Zukerman, A. E. Nicholson, and A. Bud. Towards a bayesian model for keyhole plan recognition in large domains. In *User Modeling: Proceedings of the Sixth International Conference*, pages 365–376, 1997.
2. C Beal., J. Beck, D. Westbrook, M. Atkin, and M. Cohen. Intelligent modeling of the user in an interactive environment. In *Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium*, Menlo Park, CA, 2002. AAAI Press.
3. S. Bringsjord and D. A. Ferrucci. *Artificial Intelligence and Literary Creativity*. Lawrence Erlbaum Associates, Publishers, 2000.
4. M. Cavazza, F. Charles, and S. Mead. Sex, lies, and video games: an interactive storytelling prototype. In *Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium*, Menlo Park, CA, 2002. AAAI Press.
5. K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 1997.
6. J. Laird, M. Assanie, B. Bachelor, N. Benninghoff, S. Enam, B. Jones, A. Kerfoot, C. Lauver, B. Magerko, J. Sheiman, D. Stokes, and S. Wallace. A test bed for developing intelligent synthetic characters. In *Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium*, Menlo Park, CA, 2002. AAAI Press.
7. J. E. Laird. It knows what you're going to do: Adding anticipation to a quakebot. In *In the Proceedings of Agents 2001*, pages 385–392, Montreal, CA, 2001.
8. M. Lebowitz. Story-telling as planning and learning. *Poetics*, 1985.
9. B. Magerko and J. Laird. Towards building an interactive, scenario-based training simulator. In *In the Proceedings of the Eleventh Conference on Computer-Generated Forces and Behavior Representation*, Orlando, FL, 2002.
10. M. Mateas. An oz-centric review of interactive drama and believable agents. Technical Report CMU-CS-97-156, Carnegie Mellon University, 1997.
11. M. Mateas and A. Stern. A behavior language for story-based believable agents. In *Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium*, Menlo Park, CA, 2002. AAAI Press.
12. J. R. Meehan. Tale-spin, an interactive program that writes stories. In *IJCAI*, pages 91–98, 1977.
13. A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.
14. N. M. Sgorous. Dynamic generation, management, and resolution of interactive plots. *Artificial Intelligence*, 1999.
15. P. Weyhrauch. *Guiding Interactive Drama*. PhD thesis, Carnegie Mellon, 1997.
16. R. M. Young. An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. In *Artificial Intelligence and Interactive Entertainment: Papers from the 2001 AAAI Spring Symposium*, Menlo Park, CA, 2001. AAAI Press.