Driving Interactive Drama Research through Building Complete systems

Manish Mehta, Santiago Ontañón, and **Ashwin Ram**
Cognitive Computing Lab (CCL)
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, USA
{mehtama1, santi, ashwin}@cc.gatech.edu

Abstract

Interactive drama presents one of the most challenging applications of autonomous characters, requiring characters to simultaneously engage in moment-by-moment personality-rich physical behavior, exhibit conversational competencies, and participate in a dynamically developing story arc. One way to advance the field and continue to make exciting progress is to develop building blocks needed for creating these interactive experiences that are situated in a complete system. Our research goals presented in this paper are driven by this perspective of developing a complete interactive drama architecture. Specifically, we discuss the different research challenges that we are interested in pursuing at the different building blocks required to build a complete interactive drama. We also discuss the interactive drama domain we are developing and present our initial steps in handling the research challenges.

Introduction

In recent years, there has been a growing interest in creating story based interactive systems where the player experiences a story from a first person perspective, interacts with autonomous, believable

characters and, through her interaction, influences both the characters and the overall development of the story. These systems termed Interactive drama branches off into two major research themes, one centered around research issues in autonomous, believable characters, with rich models of personalities and emotions and second drama management components that incorporate the player actions into the ongoing narrative and provide a coherent, well-formed story structure to the whole interaction. Successful future research in believable agents requires deploying such agents in completed dramas, evaluating the effectiveness of the agents in creating a compelling player experience, and using the results of the evaluation to guide future research. Our approach to carrying out research in Interactive Drama (ID) and believable agents presented in this paper is inspired by this perspective and guides our initial steps in this direction.

Research in autonomous believable agents situated in ID systems have impact in other areas as well. Autonomous characters with conversational capabilities also termed Embodied Conversational Agents (ECA) have been employed in various applications including training people (Johnson & Rickel 2000), as real estate agents (Cassell *et al.* 1999), and as educating kids about historical characters (Corradini *et al.* 2005). One way to advance the field of believable agents and Interactive Drama is to develop and study complete agents in complex domains with the ultimate goal of drawing general lessons from the specific implementations. Furthermore successful research involving fully implemented solutions, algorithms and solutions can inspire new theory and vice-versa. In this paper, we present our initial steps towards this objective through development of a complete ID experience. The inspiration behind our approach is the belief that it is important to develop and evaluate the technical approaches for believable characters in the context of a real interactive drama, not a toy domain.

One of our goals through this paper is to present the research challenges and our initial ideas on the different approaches we are pursuing for developing the different building blocks for creating a complete interactive drama experience. It is not intended to be a comprehensive list of research challenges that needs to be solved for creating a complete ID experience. Our ambition for the paper is to seek suggestions from colleagues who have been involved in similar research efforts and get suggestions on the approaches we want to undertake in different research areas.

To set the scene, we first present the significance of developing approaches in complete interactive systems followed by a discussion of different research challenges that we are pursuing. Next, we present the story domain in which we situate our research objectives and discuss our initial steps towards developing the different building blocks. We finally conclude with some future steps we plan to pursue.

Developing complete ID systems

The vision of carrying out research activities at the building blocks by developing complete systems has been emphasized by prominent AI researchers (Koller 2001; Stone 2007) as well. Peter Stone in his lecture on Computers and Thought Award at IJCAI-07 underscored the benefit of carrying out research activities at the individual bricks (individual components) and applications that require the practical unification of these various bricks within a complete cathedral (complete agents). Koller in her 2001 computers and Thought lecture presented a similar vision where she emphasized the notion that in AI, as in many communities, we have the tendency to divide a problem into well-defined pieces, and make progress on each one. But as progress is made, the problems tend to move away from each other (Koller 2001). One way of approaching and combating this issue of fragmented solutions is to create fully functional agents in complex domains and build applications that require practical unification of the various individual topics into a complete agent.

In Interactive drama and believable agents research, this vision has been realized in Facade (Mateas & Stern 2003). Our objective towards this vision is to understand and facilitate development of adaptive autonomous characters in the context of advanced interactive systems. While there has been a considerable amount of research in research issues in individual pieces required to develop a complete interactive drama experience, with the exception of Facade, there has been little effort in incorporating all of them together into a full fledged system. One of the hazards in not doing so is that research in the individual problems might be deviating too much from the real need in a real system. While developing a commercial-quality interactive drama is not one of our research objectives, our goal is to have a realistic domain that provides a setup for us to develop viable representations and techniques and a platform for realistic evaluation to guide future research.

Research Challenges

Some of the research challenges that we plan to undertake through the development of a fully fledged 3D real-time interactive story development are listed below:

- **Behavior Authoring support through behavior learning and runtime behavior modification** Hand crafted behaviors are, ultimately, software code in a complex programming language, prone to human errors. The behavior errors could be in the form of program bugs or not achieving the desired result. Another issue is that hand authoring of behavior for believable character though allows designers to craft expressive behavior for characters, but nevertheless leads to excessive authorial burden (Magerko & Laird 2003). Tools are needed to support story authors, who are typically not artificial intelligence experts, to allow them to author behaviors in an easy way.

  The initial behavior set for the game characters defined at design time results in characters that are brittle to changing world dynamics as it is difficult to imagine all the possible situation that the character would encounter in the game. Moreover, the repetitive and predictable nature of character behaviors can hamper the player experience. The key problem then, is to develop a self adapting system for these characters that is autonomously responsive to new and unforeseen circumstances keeping the author-specified personalities in mind.
- **Natural Language Generation:** Interactive domains inhabited with believable characters provide rich opportunities for Natural Language Generation (NLG). The style of output text produced by the agents should be modified based on the personality and the emotional state of the character without adding a lot of authorial burden on the author. The system should also be able to provide enough generative abilities so as to undercut the cost of developing it.
- **Drama Management and Player Modeling:** There is a growing interest in developing Drama Manager (DM) components for story based games that gently guide the player towards a story ending that exhibits a narrative arc. The application of DM approaches in real large scale games raises several research issues such as scalability, long term story planning and player modeling issues. Further Drama Management approaches require a "model" that predicts the interestingness of story arcs, so that the drama manager can plan which story arcs to pursue and provide a better player experience.

To ground our discussion, next, we present the story domain in which we are pursuing our different research goals.

Story Domain : Mystery Mansion

The interactive drama we are developing is named Mystery Mansion (MM). The story set up consists of six characters and is set up in a British mansion at the beginning of the 20th century. The player controls one of the character and is free to interact with the rest of the characters using natural language and also move freely around the house and manipulate some objects. In particular, the drama starts when two of the characters decide to celebrate an engagement party, and invite two friends to a dinner in their newly acquired mansion. The remaining two characters are the butler of the house and the father of the bride. Most of the characters have strong feelings (love or hate) for some of the other characters, and as the story unfolds the player will discover hidden relations between them. The player will take the role of one out of three possible characters and will be able to act freely in the mansion, with his actions strongly influencing the development of the game.

The story consists of four big scenes: the cocktail party, where all the characters arrive to the mansion and meet; the dinner, where some initial discussions appear; the night, where the characters go to sleep in the mansion and one (or several) murders will happen; and finally the investigation, where the survivors will try to investigate the murder. Depending on the player actions, each scene might happen in several different ways; specially the third scene, where there are about 10 possible different murders that can happen depending on what the player does. The player himself might get killed, or a combination of murders can happen. The investigation is an open ended scene where the murderer may or may not be discovered (the player himself might be implicated in the murderer and he will have to cover himself). Such a game will provide a perfect opportunity to combine all the previously mentioned research goals into a real application. In the remaining sections we provide a brief description of our initial steps towards the individual research challenges we plan to pursue through the development of Mystery Mansion .

Behavior Authoring support

Behavior authoring is the major bottleneck in interactive drama development. The standard approach for authoring autonomous characters is to hand-author behaviors or scripts that describe the character's reaction in all possible circumstances within the game world. This approach of authoring characters presents several difficulties such as difficulty in planning for all possible scenarios a character might encounter and repetitive behavior harming the believability of the characters. One of the key issue, then, is the ability to construct or adapt behavior sets.

In this section we propose two complementary approaches to deal with such problem, namely: behavior learning from demonstrations and behavior modification.

Behavior Learning from Demonstrations

Story authors are typically non AI expert, and thus defining behaviors using a programming language is not an easy task for them. They have a clear idea in mind of the behavior they want particular characters in the game to exhibit, but the barrier is encoding those ideas into actual code. An approach to address that issue is programming by demonstration, where authors actually demonstrate the behavior a character must exhibit (by controlling that character manually) and the system learns from that demonstration.

In a previous approach, we have successfully developed a system capable of learning behaviors from demonstration in a real-time strategy (RTS) domain (Ontañón *et al.* 2007), specifically in the game of Wargus (an open source implementation of the classic Warcraft 2). We use a case-based approach, where each individual behavior that is learnt is stored as a *case*. In our approach, an expert plays a game of Wargus showing a particular strategy that he wants to teach to the system. The result of that game is an execution trace that the expert can annotate by pointing out which were the goals he was pursuing with the actions he was doing. The system can then analyze such annotated trace and infer behaviors to achieve the goals the expert has annotated. For each behavior that the expert has demonstrated, the system stores the particular game state in which it was executed. The combination of a goal, a behavior and a game state defines a *case*. Thus, the system can learn several different behaviors to achieve the same goal, and store in which situation each different behavior is better.

On execution time, when the system needs to achieve a particular goal (e.g. "destroy an enemy tower" in the Wargus domain), it will look for behaviors in its learned behavior library to destroy towers, and will select the one that is most appropriate by comparing the current game state with the game states stored with each behavior, i.e. the case retrieves the most appropriate case.

The described approach has been successfully validated in the context of real-time strategy games. Thus, the next research challenge is to apply it to interactive drama, where the behaviors involve verbal and gesture actions, in addition to the motion and physical actions already considered in the framework of RTS games.

Behavior Modification

Another approach to deal with the authoring consisting of modify behaviors on an ongoing basis through an analysis of the interactions of the characters with the player and with each other, relative to the primary rhetorical objectives of the game and the individual goals of the embedded characters, thereby relieving the author of writing behaviors for all possible circumstances.

In our previous approach, we have dealt with some of the research issues listed above (Zhang *et al.* 2007). To address these issues, we have developed an approach in which agents keep track of the status of their executing behaviors, infer from their execution trace what might be wrong, and perform appropriate revisions to their behaviors. We want to push our behavior adaptation approach further through the development of Mystery Mansion domain. We are developing an architecture called Automatic Behavior Adaptation architecture (ABA) to deal with the issue of dynamic behavior modification. The architecture has three main components, the character execution component A behavior Language (ABL) (Mateas & Stern 2002), the introspection component, and the behavior modification component.

The behavior execution component, based on the ABL behavior language, manages the behavior of the non player characters in the game. To do so, the game state is read through a series of sensors that update the working memory. The ABL runtime component selects which behaviors (from the behavior library) will be executed to pursue the current character goals as a function of the content of working memory, and tracks current goals and behaviors to determine if any should spontaneously succeed or fail because of changes in the world. ABL keeps track of the current active subgoals and behaviors in the Active Behavior Tree. The introspection component is one of the key modules in the architecture. Its goal is to monitor that the character behavior is satisfying the declarative behavioral constraints for each character, and to perform blame assignment if they are not. The behavior modification component job is to receive modification requests, and modify ABL behaviors to fix constraint violations detected by the introspection module. Once a behavior has been modified, it will be pushed back into the behavior execution component (the behavior library of an executing ABL agent).

Drama Management

Drama Management approaches have been shown to improve player satisfaction in the context of interactive games (Sharma *et al.* 2007b; 2007a). However, there are several challenges that have to be solved in order to incorporate drama management into complex games such as MM . Typically, drama managers use search based techniques to decide which are the best actions to select at any moment. However, in a big complex game, the amount of possible actions available is so large, that search based techniques cannot be directly applied. Moreover, previous drama management approaches do not take into account the timing problem, i.e. in addition to decide which actions the drama manager should take, also decide *when* to take them. An exception to that, is the work done in Façade (Mateas & Stern 2003), where they use *beats* to model timing. In a real-time game such as Façade or MM the timing problem is crucial, and should be taken into account. The research topics related to drama management that MM will allow us to develop are related to a) Player preference and action modeling i.e modeling which story arcs, drama manager interventions or other game elements a particular player might enjoy and further modeling which actions each particular players are likely to execute in any given situation, b) DM action selection and timing, i.e. optimizing the search process and learn the best action instead of searching for it and deciding the proper timing for DM actions is as important as which actions to select to achieve a good game dynamics and c) Proper story representation for a complex domain like MM.

In our previous approach (Sharma *et al.* 2007b) towards drama management we have tackled the issues of player preference modeling, story representation and DM action selection. We want to move forward and start tackling the research challenges mentioned above for MM.

Natural Language Generation

In general, there is a lack of reusability in NLG tools. In our previous work (Strong *et al.* 2007), we decided to start from scratch and build a system that is reusable across game domains. Our initial system uses a set of author defined templates that can be reused across different characters and emotional states. We defined these templates as a set of sentence structures that require words and phrases from a lexicon to complete the sentence. The words and phrases that are used are dependent upon the parameters given by the output from the emotion and game event tracking. We chose to use templates for several different reasons. There is a convincing argument by van Deemter et al. (van Deemter, Krahmery, & Theunez 2003) that there is no need to distinguish between "template-based" natural language generation and "real" natural language generation. At this point, both technologies have their advantages, and vary as widely within their own class as between classes.

Our previous approach to deal with NLG issues (Strong *et al.* 2007) provides us a starting point in situating our work in an interactive domain. However, as the domain is not fully conversation centered, the opportunities to deal with issues of a real time conversation-centered domain are limited. Moving forward our work, we are extending the approach in Mystery Mansion . As described above, the story of Mystery Mansion is primarily divided into four scenes. We have currently developed hierarchical templates for verbal output generation for the first two scenes. The two scenes consist of approximately 6000 (approx 400 lines) words of dialog. The goal for creating hierarchical templates is to reuse different templates across the different scenes and characters.

Conclusion

We share the perspective of various AI researchers (Koller 2001; Stone 2007) and prominent interactive drama researchers (Mateas & Stern 2003) emphasizing the benefit of developing applications that require the practical unification of these various sub-components within a complete system. In this paper, we have presented our initial ideas on the research activities that we plan to undertake at the various building blocks situated within a complete interactive drama experience. As a future step, we are pursuing work in each of the individual components forward.

Acknowledgement

The authors would like to thank Michael Mateas, author of the ABL language and collaborator on the behavior modification project, Peng Zang for behavior modification, Manu Sharma for drama management work, and Christina Strong for the NLG system.

References

Cassell, J.; Bickmore, T.; Billinghurst, M.; Campbell, L.; Chang, K.; Vilhjalmsson, H.; and Yan, H. 1999. Embodiment in conversational interfaces: Rea. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 520–527.

Corradini, A.; Mehta, M.; Bernsen, N. O.; and Charfuelan, M. 2005. Animating an interactive conversational character for an educational game system. In *Proceedings of Conference on Intelligent User Interfaces*.

Johnson, W. L., and Rickel, J. W. 2000. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*.

Koller, D. 2001. Representation reasoning learning. In *IJCAI Computers and Thought Award talk*.

Magerko, B., and Laird, J. 2003. Building an interactive drama architecture. In *In Proceedings of TIDSE*, 226–237.

Mateas, M., and Stern, A. 2002. A behavior language for story-based believable agents. *IEEE intelligent systems and their applications* 17(4):39–47.

Mateas, M., and Stern, A. 2003. Facade: An experiment in building a fully-realized interactive drama. In *In Game Developer's Conference: Game Design Track*.

Ontañón, S.; Mishra, K.; Sugandh, N.; and Ram, A. 2007. Case-based planning and execution for real-time strategy games. In *Proc. of International Conference Case Based Reasoning*.

Sharma, M.; Mehta, M.; Ontañón, S.; and Ram, A. 2007a. Player modeling evaluation for interactive fiction. In *AIIDE 2007 Workshop on Optimizing Player Satisfaction*.

Sharma, M.; Ontañón, S.; Strong, C.; Mehta, M.; and Ram, A. 2007b. Towards player preference modeling for drama management in interactive stories. In *FLAIRS 2007*. AAAI.

Stone, P. 2007. Learning and multiagent reasoning for autonomous agents. In *IJCAI Computers and Thought Award*.

Strong, C.; Mehta, M.; Mishra, K.; Jones, A.; and Ram, A. 2007. Emotionally driven natural language generation for personality rich characters in interactive games. In *To appear in AIIDE 2007*.

van Deemter, K.; Krahmery, E.; and Theunez, M. 2003. Real vs. template-based natural language generation: a false opposition? *Ass. for Computational Linguistics*.

Zhang, P.; Mehta, M.; Mateas, M.; and Ram, A. 2007. Towards runtime behavior adaptation for embodied characters. In *Proceedings of IJCAI*.