Towards Real-time Authoring of Believable Agents in Interactive Narrative

Martin van Velsen

Institute for Creative Technologies,
13274 Fiji Way, Marina del Rey, California 90292 USA,
vvelsen@ict.usc.edu

Abstract. In this paper we present an authoring tool called Narratoria[1] that allows non-technical experts in the field of digital entertainment to create interactive narratives with 3D graphics and multimedia. Narratoria allows experts in digital entertainment to participate in the generation of story-based military training applications. Users of the tools can create story-arcs, screenplays, pedagogical goals and AI models using a single software application. Using commercial game engines, which provide direct visual output in a real-time feedback-loop, authors can view the final product as they edit.

Keywords: Authoring, Interactive Narrative, Believable Agents, Simulations, Games, Training, Machinima.

1 Introduction

The software presented in this document seeks to provide sufficient authoring capabilities to allow users to create training applications using interactive narrative without the need of programmers. Traditional film has come a long way from the days of black and white movies accompanied by live piano playing. Video games have progressed even faster but are still catching up with the narrative possibilities of film. Furthermore, those who have traditionally created story based content for film do not yet have the ability to intuitively create content for new media forms such as games and digital interactive training systems. New approaches in both interactive drama and video-game technologies pave the way for exploration of what could be called interactive films [17], but much work remains to be done to bring these capabilities to those who benefit most from these developments. Narratoria attempts to provide an integrated intuitive authoring solution by giving those users who are experts in story telling in traditional media easy to use drag and drop tools for compositing complex dramatic scenes using game technologies.

2 Authoring Interactive Narrative

In our first authoring effort, the Institute for Creative Technologies' (ICT) Leaders project [6] used a commercial game engine (Unreal Tournament™) to immerse the user in an interactive decision making game. The Leaders simulation uses cinematic cut-scenes linked together through pre-defined decision points in a story graph. Decisions are elicited from users through a Naïve Bayseian natural language classification interface, which triggers the system to progress the story to a different decision point through the display of a rendered scene in the simulation (game). In our Army oriented training application, the production started with the collection of anecdotes from Army Captains who recently returned from the field. Anecdotes were analyzed by pedagogical experts and turned into plot points or decision points. Our screenplay writers linked plot points into a story graph, where each full branch represent a story arc and nodes represent plot or decision points. Story arcs ensure a coherent and engaging experience, whereas the plot points ensure that users have to navigate decisions. Screenplay writers turned the story graph into a concrete movie script that maps decisions into engaging story arcs. Traditional scripts are linear, to be played all the way through one scene after another. The writers clustered scenes into what we call story molecules [9] ensuring scenes could lead to multiple outcomes and therefore other scenes. Each molecule then becomes self contained in that it can be played by itself but could also be combined with other molecules. This modular script in the form of a story graph was handed over to a director and our art director. A director matched the script to storyboards, whereas the art director created an inventory of characters and animations needed to express the story. Finally a team of animators in collaboration with a director composed the final game from all the available digital content stored in libraries. Collectively the authoring modules form a workflow for interactive Narrative as used by the Leaders project. Other workflows can be created by re-sequencing the existing modules or by adding new modules.

3 Authoring Believable Worlds: A Question of Language

A major hurdle in creating an authoring tool for film based interactive narrative lies in the fact that multiple experts are needed. Each of these experts might use a different expression language specific to a field. An expression language could be thought of the as way experts express the models most salient to their field. For example, software developers write code that operates on spatial X, Y and Z coordinates, whereas film directors work in terms of medium and long camera shots, etc. The software tool presented in this paper hides from the user the fact that each expert works in his or her own language. It does this by integrating a number of editing tools, such as script editors, story graph editors and timeline editors. Using this approach we ensure that the learning curve for users not intimately familiar with authoring software is gradual. Especially those interfaces such as storyboards [11], which are traditionally a paper and pencil tasks, are excellent candidates to be added to the production workflow as digital incarnations. Behind the scenes the authoring application translates the interaction into shared software structures. For example screenplays can be written using the Narratoria movie script editor module, whereas film editors use the timeline editor module. The only link between narrative

information and 3D visualization is made through anchors or markers placed in the virtual environment. They are used to place and move actors or cameras, props, etc. All other control over the world is established through code interpreting the intentions of a director who uses this to give hints to the game engine. Further character performance is achieved by code that blends animations from a library in different ways to create new gestures. Blending is accessible from the timeline module where the final visuals are composed. The game engine then acts as a digital stage where directors interact with virtual actors. By allowing software to interpret the intentions of the creative users, those users can express themselves more freely because they do not feel burdened by the demands of the computational knowledge needed to built interactive worlds. Feedback is obtained from observing the output of the visualization drivers as well as timing feedback in the timeline module. Since events in the simulation may not play when expected, the timeline gives indications as to when an action started and how long it played. Synchronization in the simulation is similar to the 'rising-edge' approach [4] where the system attempts to honor the starting times of actions.

3.1 Authoring Believable Agents

Computer controlled software agents act out the interactive script. These digital actors are under semi-autonomous control of the simulation and take performance hints from a director who plans their movements, gestures and dialog. Since the digital actors, much like their real counterparts, act from marker to marker, a bit of freedom exists for the characters to choose how to get from A to B. Actors climb hills or descent stairs to reach their destination marker and beforehand it is not known how much time this will take or what obstacles might be encountered. Additional code attempts to ensure that actions play out as they should, taking into account other events that might interfere. Another tool, assisting directors in interacting with virtual actors, is the facility of game engines to blend animations. New animations can be derived using this approach, reducing the need for large libraries of gestures and movements. Blending is accessible from the timeline module and is achieved by overlapping multiple animation actions in time.

3.2 The Cinematic Authoring Process

New projects start with a screenplay. A screenplay can be written in software such as Finaldraft™ or composed directly within the Narratoria editor. Scripts are internally segmented into scenes either by extraction or through annotation by authors. Authors depend on scenes since they represent the longest uninterrupted sequence of events and therefore form a consistent reference point throughout the authoring process. Scene data can now be accessed in the timeline editor and the story graph editor simultaneously. Those working on the story graph map out plot points and link them with story arcs using drag and drop methods. Interactions and transitions are both depicted using scenes from the script and authoring the story is achieved by assigning the appropriate scenes to either arcs or interacts that are contained in the nodes. Authors map scenes, depicting interactions with virtual actors, to classes of text input

that will be typed by users playing the game. Interact scenes can be set to also trigger arcs, which means for that specific user input maps to a decision and moves the plot to the next node. While the script is being revised and the story structure is further developed, editors familiar with non-linear film editing use the timeline to refine the cinematic content of the scenes. Some of the information comes from the script, such as dialogs, subtitles and camera shots. Other details are added manually, for instance: transitions, animations, and other aspects not found in screenplays. Editors add these visualization instructions as actions on timeline tracks. Each of the actions can be timed relative to the start of the scene and stretched to change the playback speed. Story editors try out interactions in the game and timeline editors play sequences to see how they appear on the screen. Finally the dataset compiles into a single XML file ready to be used by the game.

4 Technical Discussion

Narratoria consists of two major components: an authoring application and a visualization driver. The authoring application manages authoring modules and is written in C++ using the Qt interface development toolkit. Authoring modules load during startup and provide dedicated authoring capabilities for one particular part of a production pipeline. For example, one module manages the character bible and another presents a timeline canvas. On the visualization side we provide a driver that integrates into a number of different commercial game engines. So far we have tested this driver with Unreal Tournament™, Gamebryo™ and TVML™. Scene data can be streamed on the spot from the timeline editor to the visualization drivers by an author pressing ‘play’ for either a track, a scene or the entire screenplay.

4.1 Authoring Interoperability

With multiple authoring modules operating on the same information and with a number of displays interpreting visual data, the single most important aspect of the Narratoria architecture is its data design. Concepts from film and theatre form the basis of a global ontology, embedded in extendable core document classes [10]. Core concepts range from camera models to definitions of time and time fragments (beats) or even learning objectives for more recent interactive training applications. Heterogeneous software components can act on data that is altered or even managed by other components through an open data framework. The following characteristics contribute to the open nature of the document definitions:

- Using introspection, each part and sub part of a Narratoria data set can be extracted, identified and used by itself. For example, the XML instance of a character can be used independently from the XML containing a character bible.
- Every single part of the data set can be individually streamed into an external format such as XML, a method frequently used by similar efforts [14].

Recursive streaming allows hierarchies of data types (such as an entire document) to be streamed to an output module or visualization driver.

- Every individual Narratoria data type can be instantiated from a data stream. Given a well formed fragment of XML, that contains a valid Narratoria data type, a new instance of that type can be generated by the core system.
- The Narratoria data referencing design, described below, allows new data types to be added at runtime.

When deriving new classes, using the document and data types provided, developers are required to provide streaming facilities for novel data types. Developers are assisted in this by a collection of tools designed for this purpose. For those situations where new data types need to extend or reference other data types, three mechanisms provide means to interoperate and design new modules:

- Reference by derivation, where new capabilities are added by building on existing Narratoria classes (documents and data types). The newly created classes only need to provide a small amount of information such as a class name for introspection.
- Reference by linking, in this case new data is added to existing classes at runtime in the form of anonymous class variables. This would be analogous to adding new attributes to an object at runtime.
- Reference by association and the least intrusive of relating two pieces of data is the ability to freely associate data instances. In this case a third object, which we call the insulator object is placed in between two objects and forms the actual reference.

Using insulator objects allow users to have full control over the creation and dissolution of data references. For example, we could imagine a situation where a developer created a new document class containing a list of speech acts. Some of these speech acts might be specific to instances of characters from the character bible. In this case some character objects are connected to speech act objects by insulator objects and stored in the project document, which is designed to manage all instances of insulator objects. When the link between a speech act instance and virtual character is no longer valid the insulator object is deleted. Each insulator object contains methods for triggering a message to the user, indicating a potentially important link is about to be broken (for example a user deletes a speech act) and the action can be either allowed or dismissed.

4.2 Heterogeneous Collaboration

We described how specialized editing modules can be placed in a workflow, a workflow that can be adjusted depending on the tasks and domain. Users working in the workflow can collaborate on the same data set by dividing the data over a number of different users. We make use of the open nature of the data model for splitting up documents and allowing users to work on individual fragments. Those fragments are completely independent units, which can be integrated back into one large data set using facilities provide from within the authoring tool. For example, during the ICT Leaders project, six animators would work on scenes in the timeline editor in their own copy of the software. Finished scenes were integrated back into the final data set

by our art director and were manually checked by a director who would ensure that the separate pieces conformed to the overall vision. Finer grained extraction of data allowed animators to work on individual tracks within a scene for those scenes where many characters interact and act simultaneously. In those cases scenes assemble from individual track documents into scene documents, which are in turn combined into the final product.

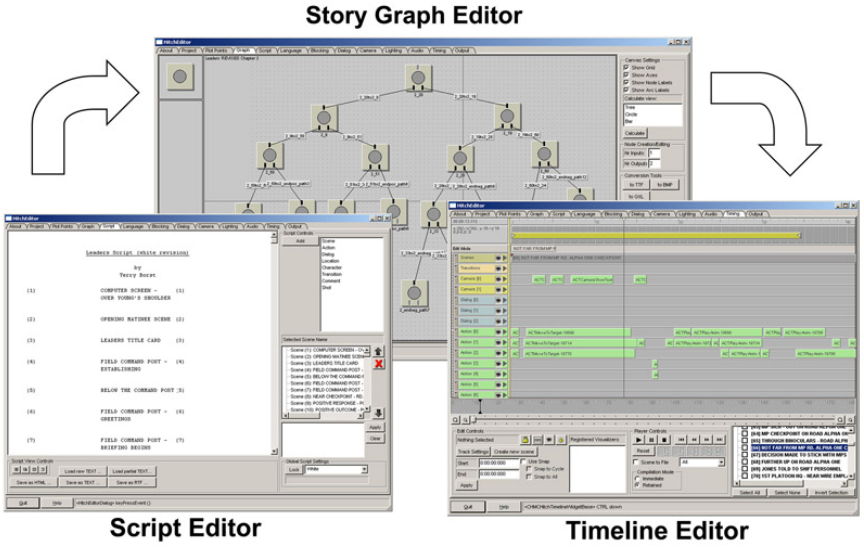

Fig. 1. Authoring Interoperability

5 Narratoria in Production, Preliminary Results

The Narratoria authoring tools have been used in numerous projects at the ICT. In the ICT Leaders project, before the introduction of authoring tools, animations were called by custom code in the game. Programmers moved the virtual actors around and had them speak lines using pre-recorded audio. Additional camera control code added means to place and move cameras. With this approach the screenplay took six months to implement. After introducing Narratoria into the pipeline the production time was reduced to three months and handed over to six animators, one art director and one cinematic director. During the last three months of production, no programmers were required to create any of the narrative content. Using the extendibility of Narratoria we recently added a pedagogical authoring module for those games that use an automated tutor. For example, in authoring the ICT's ELECT BiLAT game [8], authors use drag and drop techniques to associated learning objectives with game content that models a user's interactions. An automated tutor then uses this information to track a student's performance. Narratoria has also been applied to

ICT's Spatial Cognition project, where it generated varied scenarios for virtual characters tasked with navigating complex virtual environments [12].

6 Related Work

Similar combinations of tools exist, which take on the separate areas of screenplay writing, movie-clip sequencing and animation creation [1][13], but none integrate the tools into a configurable workflow. Some more recent applications originating in the research community do allow traditional media types to be combined with novel abstract action sequences using XML documents but these systems either do not provide an intuitive mechanism for artists to access the materials [3][4][5] or they do not allow non-traditional actions such as character animation or camera actions to be used [2][7][16].

7 Future Work

With a framework in place that is targeted towards cinema style training applications we need to address a number of factors. First of all the software needs to be adapted to different forms of interactive narrative. Second, better collaboration mechanisms are needed to support users remotely located working on data simultaneously. Finally, further research could uncover how much artists will let the simulation interpret their direction and what additional acting behaviors can be installed within the virtual actors.

Acknowledgments. The author would like to thank Roland Lesterlin and Kurosh Valanejad for supporting the development of the tools.

This research was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed herein do not necessarily reflect positions or policies of the U.S. Government; no endorsement should be inferred.

References

1. Baecker, R., Rosenthal, A. J., Friedlander, N., Smith, E., and Cohen, A.: A multimedia system for authoring motion pictures. In: Proceedings of the Fourth ACM international Conference on Multimedia (1996)
2. Bailey, B. P., Konstan, J. A., and Carlis, J. V.: DEMAIS: designing multimedia applications with interactive storyboards. In: Proceedings of the Ninth ACM international Conference on Multimedia (2001)
3. Bulterman, D. C. and Hardman, L. Structured multimedia authoring. In: ACM Transactions on Multimedia Computing, Communications and Applications (2005)

4. Drapeau, G. D.: Synchronization in the MAEstro multimedia authoring environment. In: Proceedings of the First ACM international Conference on Multimedia, (1993)
5. Gebhard, P., Kipp, M., Klesen, M., and Rist, T.: Authoring scenes for adaptive, interactive performances. In: Proceedings of the Second international Joint Conference on Autonomous Agents and Multiagent Systems (2003)
6. Gordon, A., van Lent, M., van Velsen, M., Carpenter, M., and Jhala, A.: Branching Storylines in Virtual Reality Environments for Leadership Development. Proceedings of the Innovative Applications of Artificial Intelligence Conference (IAAI) (2004)
7. Harada, K., Tanaka, E., Ogawa, R., and Hara, Y.: Anecdote: a multimedia storyboarding system with seamless authoring support. In: Proceedings of the Fourth ACM international Conference on Multimedia (1996)
8. Hill, R. W. Jr., Belanich, J., Lane, H., Core, M., Dixon, M., Forbell, E., Kim, J., Hart, J.: Pedagogically Structured Game-Based Training: Development of the ELECT BiLAT Simulation., In: Proceedings of the 25th Army Science Conference (2006)
9. Iuppa, N., Borst, B.: Story and Simulations for Serious Games. Oxford: Focal Press (2006)
10. Ivanovic, M.; Budimac, Z.: A framework for developing multimedia authoring systems: Systems, Man, and Cybernetics, 1996., IEEE International Conference on, vol.2, no., pp.1333-1338 vol.2, 14-17 Oct (1996)
11. Arnav Jhala, Curtis Rawls, Samuel Munilla, and R Michael Young,: Longboard: A Sketch Based Intelligent Storyboarding Tool for Creating Machinima.: In: Proceedings of the Florida AI Research Society Conference (2008)
12. Kim, Y., van Velsen, M., Hill, R.: Modeling Dynamic Perceptual Attention in Complex Virtual Environments. In: Proceedings of the conference on Behavior Representation in Modeling and Simulation (BRIMS) (2005)
13. Robertson, J. and Good, J.: Children's narrative development through computer game authoring. In Proceeding of the 2004 Conference on interaction Design and Children: Building A Community (2004)
14. Robertson, J. and Good, J.: Adventure Author: An Authoring Tool for 3D Virtual Reality Story Construction. In the Proceedings of the AIED-05 Workshop on Narrative Learning Environments, pp. 63-69 (2005)
15. Sung, M.; Lee, D.: A collaborative authoring system for multimedia presentation: In: Communications, 2004 IEEE International Conference on, vol.3, no., pp. 1396-1400 Vol.3, 20-24 June (2004)
16. Ueda, H., Miyatake, T., and Yoshizawa, S.: IMPACT: an interactive natural-motion-picture dedicated multimedia authoring system. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology (1991)
17. Young, R. M. and Mark, R.: Towards an Architecture for Intelligent Control of Narrative in Interactive Virtual Worlds. In: Proceedings of the International Conference on Intelligent User Interfaces (2003)