

Minstrel Remixed: User Interface and Demonstration

Brandon Tearse, Peter Mawhorter, Michael Mateas, Noah Wardrip-Fruin

Jack Baskin School of Engineering
University of California at Santa Cruz
1156 High Street, Santa Cruz, CA 95064
{batman, pmawhorter, michaelm, nwf}@soe.ucsc.edu

Abstract

This demo features a user interface for authoring stories and story fragments for use by the Minstrel Remixed story generation system. It also demonstrates Minstrel Remixed in use, allowing users to author story fragments and then have Minstrel Remixed expand these fragments and generate stories based on them. The focus is on the interface for story-fragment authoring, which exposes Minstrel's graph-of-frames knowledge representation format to the user in an interactive manner. It also exposes Minstrel Remixed's story generation capabilities as they exist currently, including the Author-Level Planning (ALP) and Transform Adapt Recall Methods (TRAM) systems.

Introduction

Minstrel, created by Scott Turner, was a landmark story generation system (Turner, 1993). This system used what we would today call a modified version of Case Based Reasoning coupled with some cleverly constructed authorial planning to generate a number of impressive stories. These stories differed significantly from their inputs and were internally cohesive. This lead Turner to argue that some aspect of Minstrel was creative. Additionally, although Minstrel's original story domain was Arthurian tales, Turner demonstrated that it could do general problem solving in other domains such as physical engineering.

Unfortunately, Turner stopped working on Minstrel in 1994 and his code was never released. Because this system has obvious potential and was inaccessible and untestable, we have rebuilt Minstrel and named the new version Minstrel Remixed. We have performed a full rational reconstruction and have investigated a number of aspects of the system including its interactivity and creativity (Tearse 2010a, 2010b, 2011).

In the past, interacting with Minstrel Remixed has been very close to programming. One would input various queries and the system would provide output in a variety of difficult to read encodings. This demonstration of our initial release of Minstrel Remixed features a new user inter-

face designed to make story construction with Minstrel Remixed easy. A video of the user interface and system in action can be found on the Minstrel Remixed website at <http://www.brandonicus.com/MinstrelRemixed/AIIDEDemo.html>

Related Work

Minstrel Remixed is similar to a number of story generation systems that exist today. Notable amongst them are its primary predecessors: Universe (Lebowitz 1984) and TaleSpin (Meehan 1977). These systems all attempted to generate narrative without simply mixing a few preconstructed sentences together in new ways. WideRuled (Skorupski 2007) should also be mentioned because it is an authorial interface on top of Universe in exactly the same way that the interface developed for Minstrel Remixed assists authoring with Minstrel. Additional notable storytelling interfaces include ScriptEase (McNaughton 2004) and Aurora (BioWare 2002).

Also important to note is Mexica (Perez 2001), the other principle story generation system that is similar to Minstrel Remixed. These two systems both attempt to generate stories without the extensive knowledge required to make them cohesive. This is done by reasoning about the world represented by a story library. Mexica is also similar in that it is being used for continuing academic improvement and experimentation.

User Interface and Demonstration Description

Minstrel represents knowledge using graphs of frames which encode stories or story fragments. Each frame has a type: "goal", "act", "state", or "belief". The different types of frames have different fields to designate things like the type of act or the actor that holds a belief. Frames are linked into a graph by typed edges, which indicate relationships (such as the fact that a particular state achieves a particular goal). Minstrel also keeps track of a list of typed nouns within a simple hierarchical ontology of types. To author stories that Minstrel can understand (which it can then use as raw material for generating new stories), these

two main elements must be manipulated. Our interface primarily displays a graph of frames, with hotkeys that allow the user to add frames, connect them with links, and edit existing frames. When editing frames, the user can fill in text values freely, but must select from the list of stored nouns for fields that denote objects (such as an "actor" field). This noun list is displayed in a side panel, with controls for adding and removing nouns. With the main area dedicated to the story graph and the side panel for noun management, the interface presents a complete story to the user and makes it immediately editable. Additionally, our interface has a panel that lists the stories in the current case library. This allows users to view other stories that Minstrel will use as raw material, and to easily switch between editing different stories.

In addition to this interface, our demonstration will also include Minstrel Remixed's story generation capabilities. It will let users select a story template and then step through the process of creating a story, using Minstrel, one action at a time. This allows users to gain an understanding of how the algorithm works to generate stories — and because the stories are editable during this process, will even allow users to interfere with or influence the generation process. Minstrel generates stories by selecting Author-Level Goals (ALGs) and pursues them using Author-Level Plans (ALPs). Our interface will show the user which goals Minstrel is selecting, and allow them to step through the execution of plans one-by-one. Additionally, Minstrel uses Transform Recall Adapt Methods (TRAMs) to do case-based reasoning when it needs to construct story content to satisfy an ALP. Our interface exposes these to the user, so that they can see which transformations were used to find a matching story fragment in the case library for each use of the TRAM system, and even which story fragment was used as the basis of the new story content generated. All of these features support our goal of rational reconstruction of the original Minstrel system. We're currently engaged in a formal analysis of the creativity of the system (Tearse 2011) and this interface facilitates this ongoing analysis.

References

- BioWare. 2002. Aurora engine toolset.
- Lebowitz, M. 1984. Creating characters in a story-telling universe. In *Poetics* 13(3), 171-94.
- McNaughton, M., Cutumisu, M., Szafron, D., Schaeffer J., Redford J., Parker D., 2004. ScriptEase: Generative Design Patterns for Computer Role-Playing Games, *19th IEEE International Conference on Automated Software Engineering*, 88-99
- Meehan, J. 1977. Tale-spin, an interactive program that writes stories. In *Proceedings of the fifth international joint conference on artificial intelligence*, 91–98. Citeseer.
- Pérez, R., Sharples, M., 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2), 119-139.

Skorupski, J., Jayapalan, L., Marquez, S., and Mateas, M. 2007. Wide ruled: A friendly interface to author-goal based story generation. *LECTURE NOTES IN COMPUTER SCIENCE*, 4871:26.

Tearse, B., Mawhorter, P. , Wardrip-fruin, N., and Mateas, M. 2011. Experimental Results from a Rational Reconstruction of Minstrel. In *Proceedings of the International Conference on Computational Creativity*

Tearse, B., Mateas, M., Wardrip-Fruin, N. 2010. MINSTREL Remixed: a rational reconstruction. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, Monterey, CA: ACM.

Tearse, B., Wardrip-Fruin, N., and Mateas, M. 2010. Minstrel Remixed : Procedurally Generating Stories. In *Proceedings of AIIDE*.

Turner, S. 1993. MINSTREL: a computer model of creativity and storytelling. *Dissertation*, University of California at Los Angeles Los Angeles, CA, USA.