Generative Platform: An Extensible Audiovisual Narrative Development and Deployment Architecture

Jason M. Reizner, BMA
Vancouver, British Columbia, Canada
www.reizner.org
e-mail: info@reizner.org

Abstract

A proof-of-concept system is built upon the Generative Platform environment (GP), a lightweight and non-exclusionary architecture for developing and deploying dynamic audiovisual narratives, both online and in situ. GP builds upon the near ubiquity of the Flash runtime and exploits its current position as the de facto standard for online video distribution, extending input support for the vast majority of video content sources found on the web. Consequently, narratives generated by GP are executable on nearly 99% of internet-connected personal computers in use today, ensuring consistent content accessibility. Generative Platform delegates functionality between three linked modules: a source data parser, a generative logic sequencer and event handling system, and a multithreaded audiovisual playback subsystem. XML support is inherent to GP, enabling the capability to import and export a variety of data sources including XSMP, RSS feeds and automated archival search results. Object-oriented iterative sequencing logic is programmed using standard ECMAScript-syntax. Video compositing and playback are designed to minimize processing and network overhead, allowing higher framerates and more universal user experiences.

1. Introduction

From its conception, Generative Platform has been built to accommodate the vast stores of digital video data available online and in formal archives around the world. The ability to approach these resources and reliably transform them into rational source material for subsequent creative endeavors will inevitably transform the ways in which time-based narratives are both crafted and consumed. By designing a system that supports the simultaneous processing of video data and associated metadata, it is possible to implement comprehensive narrative ontologies that are otherwise impossible within existing single-scheme multimedia frameworks. The capability of leveraging linked data, such as rhetorical annotations and semantic search results, enables the creation of structured user experiences that transcend the trope of hypervideo simply as collage. Espousing this paradigm, the objective is to further explore the grey area between conventional archivism and collective distributed content creation while facilitating the construction of new methodologies for working with digital time-based media.

The construction and release of a robust, extensible and well-documented open toolset such as GP enables other media practitioners who desire to develop strong generative narrative concepts to do so without having to first establish an entire programming framework. Through the modular componentization of the technical functions necessary to support these narratives, creators are free to devote their focus and energy to the realization of a specific conceptual vision or research objective, rather than to the design and development of bespoke software infrastructure. Consequently, the establishment of a unified development architecture facilitates the unhindered exchange of methodologies and techniques between practitioners, and creates the potential for collaborative communities between artists, programmers, researchers and archivists.

2. Related Work

Countless approaches to the production of generative audiovisual narratives have been previously devised and published, however, most of these systems have been developed and purpose-built to execute a single narrative ontology. Typically, once a conceptual agenda has been established, the production of a generative narrative first requires several processes, including the procurement, organization and indexing of audiovisual source material, and the construction of the processing mechanisms necessary to execute the system.[1] GP is unique in its design approach, in so far as it streamlines the current course of development by providing the foundations of these mechanisms for managing custom datasets, as well as interfacing with pre-existing external video and metadata sources. By minimizing the need for additional software support, the focus of development can be shifted towards the realization of more elaborate conceptual frameworks and richer narrative experiences.

3. Technical Overview

Generative Platform is designed around a core set of objectives: accessibility, consistency, efficiency and extensibility. These objectives are reflected in the both the user and developer paradigms that underlie the architecture, as well as in the technical requirements necessary to support it. As a result, the overwhelming majority of internet-connected personal computers in operation today are capable of executing narratives built with GP. Although the Flash runtime upon which GP is based is proprietary, its near ubiquity and current market position as the de facto standard in web-based video has left few viable alternatives, either open-source or closed. A synergistic benefit of this dependency, however, is realized by leveraging the existing skillsets and widespread technical knowledge of practitioners already well versed in Flash development practices.

The functional underpinnings of the architecture are divided into three tightly linked components, each playing an integral role in facilitating the production of a generative narrative. An overview of the functional relationships between these components and the broader narrative ecosystem is elaborated below, in figure 1.

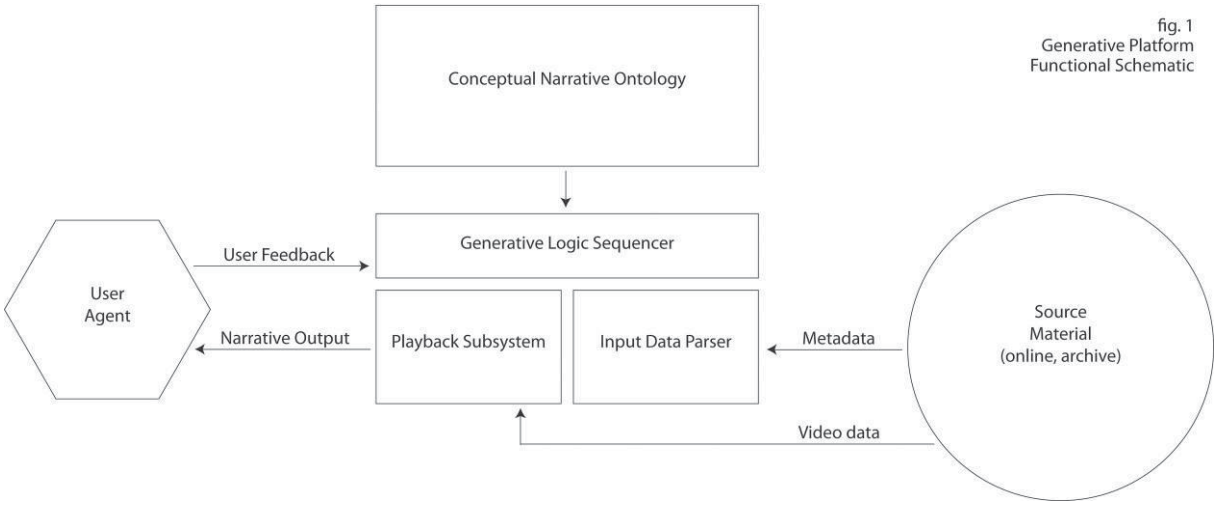

fig. 1
Generative Platform
Functional Schematic

3.1 Generative Logic Sequencer

As the primary control structure within the Generative Platform architecture, the Generative Logic Sequencer is both the procedural heart of the narrative environment and primary site for the programmatic definition and structuring of the Conceptual Narrative Ontology, or the models that specify all “concepts or objects, their attributes and inter-relationships”[2] within the experiences themselves. Evaluating object-oriented routines formalized in standard ECMAScript syntax, this module mediates the linked data accrued by the Input Data Parser, and provides all display and composition decision guidance to the Playback Subsystem.

3.2 Input Data Parser

Responsible for the ingest of linked metadata, the Input Data Parser is vital to managing and articulating the datasets that bring order and relevance to the video content being processed.[3] This module is capable of handling a variety of frequently used data formats, including XSMP, RSS feeds, and standard Dublin Core metadata. Additionally, support for other XML-based metadata schema can be added as they are encountered, enabling connectivity with a multitude of database systems.

3.3 Playback Subsystem

All audiovisual content is loaded, queued, rendered and displayed by the Playback subsystem. Working in tandem with the Input Data Parser, this module provides a unified, scalable interface to the audiovisual capabilities of the Flash runtime, and a centralized clearinghouse for the orderly request of bandwidth-intensive video content. GP supports the usage of installed Flash runtimes starting from version 7, allowing developers to target specific user agents at their discretion, based on their playback requirements. As such, support for a range of H.264 formats is available to developers targeting the most recent versions of the runtime, while also preserving compatibility for users with older installations.

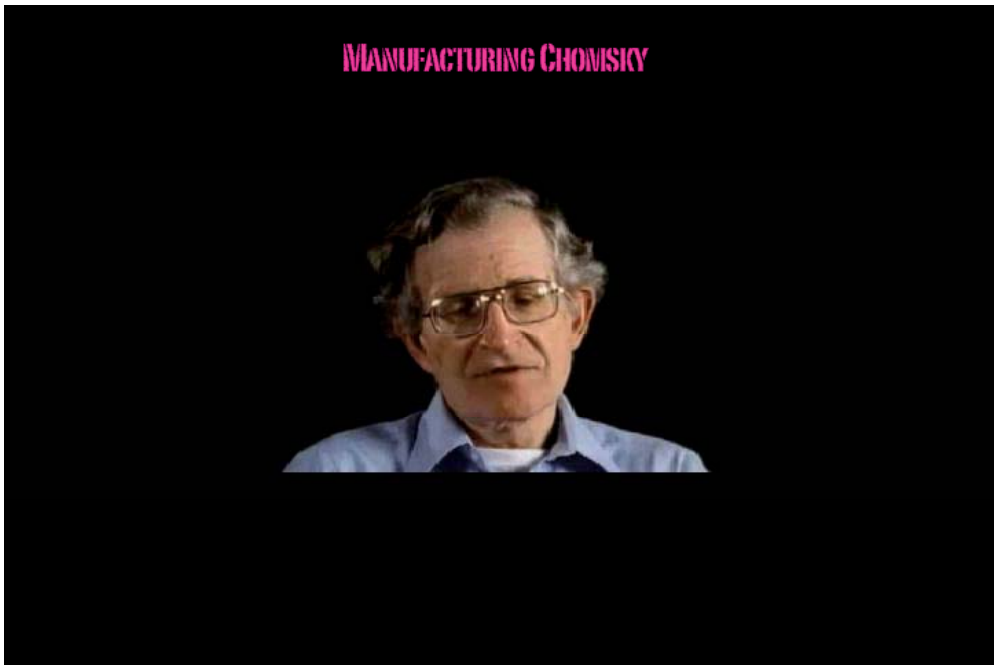

fig. 2: *image of "Manufacturing Chomsky", built on GP*
http://reizner.org/chom

4. References

[1] Laskari, Iro et al, "Generative System for the Synthesis of Audiovisual Narrative", National & Kapodistrian University of Athens, 2008

[2] Buckingham Shum, S. et al, "ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse", Knowledge Media Institute, The Open University, Milton Keynes, 2000

[3] Bocconi, S. et al, "Using rhetorical annotations for generating video documentaries", Centrum voor Wiskunde en Informatica, Amsterdam, 2005