White Paper Report

Report ID: 98476

Application Number: HD5044108

Project Director: Douglas Reside (dreside@umd.edu)

Institution: University of Maryland, College Park

Reporting Period: 9/1/2008-8/31/2010

Report Due: 11/30/2010

Date Submitted: 8/24/2010

Music Theatre Online Final Report

Music Theatre Online (originally the Electronic Broadway Project) was launched in March of 2010 to coincide with the launch of the Glory Days cast album by Ghostlight Records. The announcement of the site was frequently tweeted and picked up by the web-based crowd-edited digital humanities publication, DH Now. The annotation code has been adapted by the Chymistry of Issaac Newton project at Indiana University, the audio linking tool has been used for a Danny Kaye web exhibit currently being developed at the Library of Congress, and the project was the basis for a new grant application by the project staff to Scholarly Editions Program at the NEH. Although that application was not funded, grant reviewers acknowledged that the project staff had demonstrated technical expertise and remarked that "It is attractive that the project materials are to be made available free of charge on the web in the Music Theater Online archive." The site was widely praised by the Twitter-verse, and has been visited by users from around the globe (including England, Australia, and Japan).

We selected *Glory Days* as our initial title because we were able to get the rights and because, during its run in Washington D.C. it was very well reviewed and showed a great deal of promise. When we submitted our grant application, it had just been announced that the musical was headed to Broadway. Unfortunately, the New York production was less well received and the show closed after a single performance. Perhaps partly as a result, actual use of the site has thus far been relatively limited. According to Google analytics, in an average month there are less than 25 unique visits to the site. In part, we believe this is because the content currently in the archive is of limited interest to the scholarly audience for whom the archive is intended. Had the show run longer, we suspect interest in the site would have been much greater. Nonethless, the infrastructure we have built is content agnostic, and new titles could easily be added if funding could be obtained. We are currently collecting content from several out-of-copyright musicals including *The Black Crook* and Jerome Kern's *Sally* which we hope to add to the site in the near future. Additionally, Northwestern theatre scholar Tracy Davis has shown interest in editing the British musical, Dorothy, as a future contribution to the site. We feel we have successfully built a usable interface for this exceptionally multimodal artform, and believe when content of broader interest is added to the archive, use and reception will improve.

Process

MTO was built using the Yahoo User Interface (YUI) JavaScript framework, a very small amount of PHP, QuickTime for music playback, FlowPlayer for Video Playback, and TEI-compliant XML to encode the texts. Music Theatre Online is one of the very few TEI-based projects that attempts to link text, music, video, and image to create an electronic edition of a single work. To accomplish this we needed a way to link the verse line to the filename of the audio file and the timestamp for the moment where the line is sung. There are several ways this could have been handled. The synchronized multimedia integration language (SMIL) was our first choice. It is expressed in XML, like TEI, and has been used by several other TEI projects (including a demonstration with stylesheets that transform among METS, TEI, and SMIL by NYU's digital library. Ultimately, this solution seemed a little too complicated for our purposes. SMIL seemed best suited for presentations where the clock was assumed to begin running at the start of presentation and to continue ticking as long as the user was watching. We wanted to place the text at the front of the user experience and only call up the multimedia if the user specifically requested it. Further, we were reluctant to force our theater graduate student who previously had no XML encoding experience to learn several schemas, especially as it seemed to us that TEI had everything we needed.

To begin, I wrote a script that assigned a unique XML:ID attribute to every tag in the TEI document which didn't have such an attribute already. When working with XML, whether using XSLTs or JavaScript DOM functions, I frequently find myself wishing I could address a particular tag directly without crawling down through child and sibling nodes or setting up a complicated X-Path query. Although this process clearly adds slightly to the size of the text, we feel the power it allowed us was more than worth an extra couple of kilobytes. I then created a "timeline" of each song with each timestamp represented by a "when" with a unique XML:id. Finally, I created two "linkgrps." The first linked the "l" tag encoding a line of the song to the "when" tag in the timeline (thus linking timestamp to line) and the second linked the id of the timeline to the filename of the song. This last move is the one place where I believe I broke the schema. The link tag should probably point to another id which would in turn contain the external href pointer to the file itself. However, there did not seem to be a good, standard way to do this, and so I decided to slightly break the model rather than try to make the TEI truly valid. Because this break point is localized in one, small area of the file, it would be an easy matter to correct this problem should a better way emerge. When the XSLT transforms the sung lines into HTML, it adds an onclick event which fires a function that looks up the song and timestamp in an array which is generated when the XML is loaded.

We also wished to track the development of the script from the first drafts through the rehearsals and readings to the final Broadway version, and, through the use of a web-based collation tool, compare any one version of the script to another. The Versioning Machine is a commonly used interface for displaying versions, but it requires the differences to be encoded in the TEI itself. We preferred to keep a one to one match between source document and TEI file, allowing each file to be agnostic of all others in the archive.

In order to automatically collate the differences between the files, I modified the source code of the popular implementation of the 1976 diff algorithm, JUXTA (developed by the Nineteenth Century Scholarship Online (NINES). JUXTA can be used to generate sophisticated comparisons of multiple texts, but it does not easily export these comparisons to a format that can be viewed on the web. I therefore modified the output of the Diff class in Juxta source to output a set of XML files which collated every possible pair of files in a given directory. I then wrote a PHP script to generate HTML representations of each of these collations, each with various JavaScript onclick events which get picked up and processed by the interface.

Finally, the Music Theatre Online interface also includes an annotation tool that uses standoff markup to allow users to notate any region of the text--even across existing TEI tags. As in the case of the audio links, the functionality was made possible by the presence of a unique xml:id for every tag. Because of this, any point in the document can be described by three points of reference, the parentID, the child of that parent in which the point occurs, and the offset within that child. If a note is modified or deleted, only the other inserted notes in that parent node need be modified.

These tag sets can be downloaded and shared with other users, allowing, for instance, a stage manager of a production of *Glory Days* to take notes during rehearsals to be shared with the cast at the end of the day. Unlike many similar projects, however, none of the annotations are stored on the project server. While limiting the possibilities for aggregating user generated content, this approach does permit users to maintain complete control of their work and absolves MITH of the responsibility for maintaining an annotations server. When the user saves an annotation set, a JSON string is generating which identifies all of the insertion points for the annotations and the user-generated note. This JSON string is attached to a JavaScript variable in a very short HTML file which is sent to the user for download. When the user opens this HTML file, the page automatically sends form data to the MTO server that then applies the annotations to the text. In the near future we plan to integrate the code from the Ajax XML Encoder (AXE) to allow users to annotate images, video and audio files as well.

Future Work

MTO is the final MITH project to have used the Yahoo User Interface JavaScript. Starting in 2010 all of our new work has been designed using the somewhat more efficient and much more widely adopted jQuery, and, in a future phase, I would very much like to migrate all of the code to this framework. Similarly, MTO uses a set of multimedia playback plugins that we no longer tend to use at MITH. QuickTime, for instance, is not generally available on Linux browsers, and as a proprietary closed source plugin cannot be easily or legally migrated to new platforms. FlowPlayer is much more open (release under the GPL license), but, in the free version, includes a company watermark that cannot be removed. MITH is currently using Projekktor, another GPL-licensed plugin that defaults to HTML5 native video and audio playback where available. A new version of MTO would use this code.

The most important piece missing from our archive at present is a way of representing and preserving dance.   For the next phase of the project we hope to develop an XML schema for a Dance Encoding Initiative. Just as TEI provides tags both for linguistic information as well as for structural divisions and semantic content of a text, so metadata for dance should be expressive enough to classify not just gross movement but also commonly performed emotions or concepts. It might be interesting, for example, to compare and contrast the ways in which Jerome Robbins and Twyla Tharp dance an idea like “loss.” If the tag was applied to video records (or perhaps even 3-dimensional visualizations of data from motion capture technology) in the appropriate places, it would be possible for a scholar to easily search the video archive and make this comparison.

There are, of course, several problems with such an approach. First among them is the assumption that the terpsichorean can always be translated to the linguistic—there is not, perhaps, a stylesheet for transforming dance to language. However, this is a problem inherent in dance scholarship at present. As humanities scholars we communicate in spoken and written language and, not just in dance, must continually find ways to articulate those ideas, feelings, and phenomena for which words are woefully inadequate. For scholars who are communicating about dance using words, it seems appropriate to provide a means for bridging the gap between movement and language.

Conclusion

Music Theatre Online has thus far been very successful as a prototype interface and code base. The prototype has more than accomplished what we set out to do, but we hope to find additional funding to make the content as exciting as the presentation.