Using a Single Archive Audio File Format for Archive, Discovery and Display.

John Sarnowski; The ResCarta Foundation; Onalaska, WI, USA

Abstract

In 2012, the ResCarta Foundation met with staff at the Library of Congress and had conversations with George Blood Audio of Philadelphia concerning the adoption of an audio file format for long term preservation, discovery and display. There was consensus that using the existing Broadcast wave file format with metadata embedded into the axml chunk would be a starting point. The ResCarta Foundation has released a version of the ResCarta Toolkit which can create Broadcast wave files with embedded METS formatted metadata in the axml chunk and uses marker chunks to provide full transcription with time offsets.

This file format will allow a single digital file to carry the associated technical, administrative and descriptive metadata and allows for full text search within transcriptions. The use of such a file format in oral history projects will allow for finer granularity in the use of archived digital audio files.

The ResCarta software is utilized around the world for textual and pictorial digital objects. With the addition of audio reformatting, it can create raw transcriptions of audio files and has a full, built-in audio transcription editor. The associated web application allows for full text search within transcriptions as it does for textual digital objects. It displays the location of each searched term within a graphical audio waveform.

Introduction

For over a decade, the ResCarta Foundation has assisted organizations in converting source textual material to digital format. Books, microfilm, photographs and newspapers have been added to a standardized image format with embedded Library of Congress MODS [1] metadata.

The digitized paper and film images are stored in TIFF files with the descriptive, administrative, technical, and textual metadata elements stored in open, published TIFF tags.

In order to create standardized directory structures, file names, embedded and external metadata and checksum values, the Foundation created an open source software suite to manage the conversion. This included software for metadata creation and data conversion. To further the reusability of the created digital objects, the Foundation added a collection management software tool for gathering together digital objects and augmenting the established metadata. There also was a need for a textual metadata editor since the use of unedited optical character recognition (OCR) output in the creation of PDF image and text files had become commonplace.

With the creation of a common storage format came the need for software to assist in discovery, display and maintenance of the archive. An indexer was added to the toolkit to create open Lucene™ indexes to the metadata, a web application running under Apache Tomcat™ was written to expose the archive quality images to local networks or the World Wide Web and a checksum validation program was developed to provide a method to test the archive's integrity.

Audio and video formats were not supported in the original releases of the toolkit. The Foundation was waiting on work being done by the Corporation for Public Broadcasting and National Public Radio on standardization of multimedia formats. When it became evident that there was no consensus on an audio or video archive format [2], the Foundation decided to start with known acceptable formats for audio and to create a software tool for creating an open format for inclusion in its ResCarta Toolkit.

How ResCarta Thinks

The Foundation attempts to reduce the complexity of known formats by selecting subsets of standards to which it can adhere and producing computer software to assist others in creating standard compliant storage systems for their digital objects.

Since most common computer systems use raster based video monitors and printers, it seemed obvious to the Foundation to select a raster image format for storage of image based materials like photographs and printed text. The selection of uncompressed or G4 compressed TIFF for this storage component was well supported by standard practices.

Metadata in all forms (administrative, descriptive, technical and textual) should be written in a known format outside of a proprietary database and best to be embedded within the raster storage file.

Rather than allow for the complexities of a design-it-yourself data structure, the Foundation looked to existing best practices which led to a choice between simple Dublin Core (DC) or the extensive Metadata Object Description System (MODS) from the Library of Congress. The Foundation settled on MODS for metadata storage because its finer granularity can be converted directly to DC while the less eloquent DC can not be mapped directly back into MODS.

For textual materials the Foundation stores location and font information separately from the UTF8 text components. This allows for other systems to read the textual content of the digital object without the need to parse the extraneous location and font information.

All metadata for each finest granular digital object is stored within the TIFF header tags.

An external Metadata Encoding and Transmission Standard (METS) xml file stores the structural metadata and checksums for multi-image digital objects and repeats the MODS information stored in each component raster image file. The Foundation's practice of using best practices to create the basic ResCarta metadata was followed when considering standards for audio archives.

An Audio Archive Format

Keeping the design in line with the structure of the TIFF storage system, the audio storage system needed to have a well documented, widely used file format capable of carrying metadata within itself.

The Broadcast Wave format (BWF) seemed the likely choice. After conversations with staff of the Library of Congress, as well as staff at George Blood Audio and others, the Foundation staff set out to complete the audio package using BWF.

The first step was to write a BWF file from a Microsoft Wav file and verify that the BWF was compliant with other tests and contained the required BEXT metadata elements . This was done successfully.

Embedding Metadata in a Standard Format

The next step was to write a MODS xml formatted file into the axml chunk of the BWF and verify that it could be read back. We chose BWFmetaedit [3] from the Federal Agencies Digitization Guidelines Initiative (FADGI) as a logical test tool.

Our first test implementation resulted with the data in the axml chunk declared missing. We reviewed the BWF specifications and discovered a bug in the BWFmetaedit. The bug involved the use of "aXML" instead of "axml" for the axml chunk ID. There were a few other bugs in BWFmetaedit concerning the use of null padding and the MD5 computing was wrong on 64-bit architecture and big endian architectures. These errors were reported to Library of Congress staff who later released version 1.2.0 that validated our output. Having proven the ability of our code to create acceptable BWF files with standardized technical, descriptive and administrative metadata, we proceeded to work on matching the capabilities of our digital audio objects to existing digital text and image objects.

Transcription Capability

With a valid BWF file containing a MODS xml metadata record in the axml chunk, we set out to add the capability of storing textual metadata in the file. Our first thought was to use cue chunks for this. As we looked into their use we agreed with the authors of EBU – TECH 3306 [4] in their statement :

> "• The existing cue chunk is functional only for the first (lowest) 4 Gbyte of audio data in an RF64 file, because the legacy cue chunk uses 32 bit addressing.
>
> • Experience has shown that the definition of the RIFF/WAVE cue chunk has been interpreted ambiguously, giving rise to some developers implementing marker functionality in an improper way in their applications.
>
> • Software developers have to handle markers differently, depending on whether linear or compressed audio is the payload, which adversely affects simplicity and accuracy of the resulting code.
>
> • Labels are not stored in the cue chunk, but in a different, label chunk, which is an unnecessary complication."

By using the MBWF/RF64 implementation of marker chunks we avoided the need to recode for files that may become larger than four gigabytes and the complexity of using links to label chunks. The result closely resembled the use of TIFF tags for text in TIFF image files in that one could dump the transcription textual data from the file easily.

The Toolkit

Once the writing of a standard audio file with embedded metadata (BEXT, MODS, and Transcription) was completed, we added the code to our existing ResCarta Toolkit. This code is implemented in the Data Conversion Tool (DCT), which takes TIFF, JPG, PDF, and WAV files and converts them to the appropriate archive format with embedded metadata.

SPHINX for Transcription Creation

With image formats like PDF that contain text generated by word processors or OCR, the DCT rips the word, font and word location from the original file and embeds the textual metadata into the TIFF archive file.

When an audio wav file contains spoken words, the DCT creates a raw transcription of the contents. We use the CMU SPHINX toolkit [5] turned on by a simple check box on the DCT window.

Audio Transcription Editor

Using the raw transcription generated either from OCR for images or from SPHINX for audio files may produce poor results from poor source materials: garbage in garbage out (GIGO). Within the ResCarta toolkit there is a Textual Metadata Editor (TME) used to edit OCRed text. Similar in function to this is the newly developed Audio Transcription Editor (ATE). The ATE allows a user to visually move through an audio transcription while listening to the original recording. There is a built-in spell checker that can assist with the correction. By using the ATE in conjunction with the SPHINX transcription, it
is possible to extend the usefulness of audio files by extending the possibility of discovery.

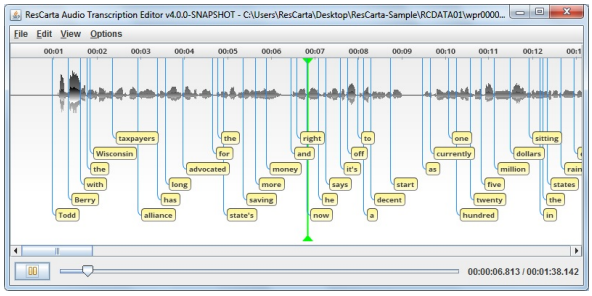

***Figure 1**. ATE Screenshot Showing Word Blocks and Time Values*

Discovery

The raw or fully edited transcription stored in the marker chunks along with the contents of the MODS metadata in the axml chunk can be incorporated into a unified Lucene index. The existing ResCarta Indexer was expanded to include audio objects for quick location of individual words or phrases within an audio stream.

Having a known standardized directory structure and naming scheme for image based and audio based digital objects makes locating a specific object trivial.

The ResCarta Toolkit also provides a Collection Management (CMT) tool for gathering like items into separate collections. The CMT allows for augmenting existing digital object metadata at the collection level. The CMT writes out a collection level Metadata Encoding and Transmission Standard (METS) xml file. This tool can also write out Open Archives Initiative (OAI) Dublin Core format or simple Dublin Core metadata of objects in the collections. These formats can be used in an Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) Data Provider to expose the metadata to the Internet.

Display

In order to serve the archive audio content to the Web from a single server, we have to overcome the current desperate support of audio formats among the current stock of web browsers [6]. The Foundation provides an Apache Tomcat™ web application that reads the archive directly as well as provides audio playback for the current crop of web browsers.

Using full resolution archive TIFF images as a source the ResCarta-Web application delivers JPEG to browsers from the server on the fly. Additional use of thumbnail images can be generated by the application on demand.

Serving the BWF audio content at this time is done through the use of an OGG file that is generated at the time of data conversion. Since neither WAV nor OGG are natively supported by all browsers, the ResCarta-Web application handles playback after recognition of the browser.

***Figure 2**. ResCarta-Web Audio Display with Highlighting.*

Text highlighting during searches has become commonplace and an expected feature for search engines. By storing the transcription words and time offsets, the same can be done for audio searching as show in the screenshot above of a search for the term 'Prilosec'.

Searching the abstract of this radio show (seen in Figure 2.) or any of the descriptive metadata elements would not reveal that this show contained over twenty references to the use of Prilosec™. Without the marker chunks a text transcription of the show would force a researcher to listen to the hour long show in real time to find the use of the term. But with ResCarta's tool, the words are found and highlighted.

Maintenance

The BWF audio file contains a checksum value for the data chunk within the file. This is used to determine if there is any bit deterioration or change to the original data chunk. The Data Conversion tool also writes a checksum of the entire BWF file into an external METS xml file. This file can be used to check the validity of the metadata stored within the BWF file. A Checksum Verification Tool (CVT) is also included in the software toolkit for this purpose.

Specification Confusion (Tech Talk)

During the production of the code to produce an open standard audio format, that contains the audio stream, various metadata formats and full transcription capability, there are times when the existing specifications are unclear or can be interpreted differently. In trying to create files that adhere to the FADGI guidelines there were times when we had to choose between the BWF specification and the FADGI. As with the cue chunk in BWF there was some confusion about the way the specification depicted the marker chunk, which could also be interpreted differently.

The RFC 4122 UUID specification defines the following structure for UUID values:

```
typedef struct {
   unsigned32 time_low;
   unsigned16 time_mid;
   unsigned16 time_hi_and_version;
   unsigned8 clock_seq_hi_and_reserved;
   unsigned8 clock_seq_low;
   byte node[6];
} uuid_t;
```

Given the structure defined the UUID specification and the little endian byte order of a RIFF file, the GUID ca61a558-4a55-4a77-af72-f17c2fb933c9 would be written as follows:

Table 1. UUID example

time_low	58 a5 61 ca
time_mid	55 4a
time_hi_and_version	77 4a
clock_seq_hi_and_reserved	af
clock_seq_low	72
node	f1 7c 2f b9 33 c9

The EBU Tech 3306-2009 RF64 specification defines the following GUID structure:

```
struct Guid
 {
   unsigned int32 data1;
   unsigned int16 data2;
   unsigned int16 data3;
   unsigned int32 data4;
   unsigned int32 data5;
};
```

Given the structure defined the RF64 specification and the little endian byte order of a RIFF file, the GUID ca61a558-4a55-4a77-af72-f17c2fb933c9 would be written as follows:

Table 2. RF64 GUID example

data1	58 a5 61 ca
data2	55 4a
data3	77 4a
data4	7c f1 72 af
data5	c9 33 b9 2f

The difference between these two forms lies in the order of the last 8 bytes of data. Commonly the last 8 bytes of a GUID are always written in the same order regardless of endianness. However, the RF64 specification seems to suggest a different approach.

We have chosen to follow the RF64 specification literally as in the second example.

It will in any case be reread into the original GUID by our software and any other software implementation will derive a unique GUID, which in most cases will be unique.

Besides the few areas of concern with existing specification, we feel that the format of the BWF audio archive file written by the ResCarta Toolkit will serve anyone wishing to store digital audio safely. We feel that by providing an open source software suite to enable the use of Broadcast WAV Files for audio archiving can assist in the creation of exchangeable and reusable archives that have the potential for a more granular search of the spoken word.

Next steps.

Although the RF64 allows for a BWF-compatible multichannel file format enabling file sizes to exceed four gigabytes at various bit rates, our initial release supports only one or two channel 16bit, 44.1khz WAV files as source audio. We intend to support all variations of channel counts and bit depths in future releases. In any case, we felt it best to release the code with support for 44.1khz and receive comment prior to expanding the code set to be all-inclusive at this time.

References

[1] MODS (Metadata Object Description Standard) http://www.loc.gov/standards/mets/mets-schemadocs.html

[2] Preserving Digital Public Television, pg. 14. (2010) http://www.digitalpreservation.gov/partners/documents/pdpt_finalreport0610.pdf

[3] Federal Agencies Digitization Guidelines Initiative http://www.digitizationguidelines.gov/guidelines/digitize-embedding.html

[4] MBWF/RF64:An extended File Format for Audio, pg.10. (2009) http://tech.ebu.ch/docs/tech/tech3306-2009.pdf

[5] CMU SPHINX Open Source Toolkit For Speech Recognition http://cmusphinx.sourceforge.net/

[6] HTML5 Audio, Wikipedia, the free encyclopedia. 02-Feb-2013. http://en.wikipedia.org/wiki/HTML5_Audio

Author Biography

John Sarnowski has a BS in theology from Saint Mary's University of Minnesota. (1970) and has over 20 years' experience in building digital collections. He was responsible for creating millions of digital objects for learned societies, libraries and major corporations as the director of Imaging Products at Northern Micrographics. He is a member of ALA, WLA and IS&T.