A Standard Audio Encapsulation Method

Homayoon Beigi[1]
Recognition Technologies, Inc.
3616 Edgehill Road
Yorktown Heights, NY 10598, USA
beigi@RecognitionTechnologies.com

Judith A. Markowitz[2]
J. Markowitz Consultants
5801 North Sheridan Road, Suite 19A
Chicago, IL 60660, USA
judith@JMarkowitz.com

Abstract

This is a position paper on the following topic listed in the call for contributions: *Can audio formats be normalized for interoperability and conformance testing? How to determine that normalization?* The position regarding the first question is that audio formats can be normalized. The response to the second question is a proposal that supports the representation of spoken audio using only existing standards. The proposal is part of a draft standard developed by the the M1 (biometrics) committee of the computing technology sector (InterNational Committee for Information Technology Standards) of the American National Standards Institute (**ANSI/INCITS**) that is undergoing the public review process.

1 Introduction

Speaker Identification and Verification (**SIV**) refers to one of the fundamental divisions of speech processing that can be extended to include classification, segmentation, tracking and detection of speakers.[1] Despite the importance of these operations, there is remarkably little work on SIV standards.[8] Consequently, the anticipated SIV module for VoiceXML 3.0 represents a significant development. The question of whether to standardize audio formats in VoiceXML has been a hotly-debated topic within the Voice Browser Working Group (**VBWG**) of the W3C. Considerations, such as the plethora of audio formats and the popularity of some proprietary formats argue against normalizing audio formats in VoiceXML.

This paper extends that debate to development of the SIV module for VoiceXML 3.0. The position taken is that normalization of audio formats is essential for SIV because of the need for interoperability and data sharing. Data sharing (and, hence, interoperability) is necessary for national security,law enforcement, and anti-terrorism activities. Important commercial correlates of those activities are the growing use of *black lists* of fraudsters by financial services, telecommunications and other industries and of repositories of authorized users that support global commerce. Furthermore, normalization of audio formats facilitates migration of applications to new engines, which is one of the fundamental rationales for using standards.

The proposal in this paper utilizes widely-used, open standards. It is designed to support all possible audio interchange while steering away from proprietary formats. It was incorporated into a draft standard of the M1 committee of **ANSI/INCITS** that is currently undergoing public review.[7] It was also submitted to **ISO/JTC1 SC37** (biometrics) as a U.S. contribution to the Data Interchange Format for Voice.[5]

2 Audio Encapsulation Standardization

The main idea of this proposal is to be able to use exiting audio formats by bringing them under one cover so that different needs of the Speaker Biometrics community are met without having to resort to using proprietary formats. Considering the various scenarios for audio interchange, three different goals are most prevalent. Table 1 presents these scenarios and the proposed audio format(s) for each case. This section describes the different cases in more detail.

2.1 The Uncompressed Non-Streaming Case

Linear Pulse Code Modulation (LPCM) is the method of choice for this kind of audio representation.[1] There is no compression involved in either the amplitude domain or the frequency domain. The bare-minimum information needed in the header for this format is the number of channels, the sampling rate and the sample size (in bits). Table 3 includes this header data and some additional information.

[1] Homayoon Beigi is the President of Recognition Technologies, Inc. and an Adjunct Professor of Mechanical Engineering at Columbia University
[2] Judith A. Markowitz is the President and Founder of J. Markowitz Consultants

Quality	Format
Lossless	Linear PCM (LPCM)
Amplitude Compression	μ-law (PCMU) and *A*-law (PCMA)
Aggressive variable bit-rate compression	OGG Vorbis
Streaming	OGG Media Stream

Table 1: Audio Interchange Scenarios

Macro	Value
AF_FORMAT_UNKNOWN	0x0000
AF_FORMAT_LINEAR_PCM	0x0001
AF_FORMAT_MULAW	0x0002
AF_FORMAT_ALAW	0x0003
AF_FORMAT_OGG_VORBIS	0x0004
AF_FORMAT_OGG_STREAM	0x1000

Table 2: Macros

Microsoft WAV is not included because it is not a format; it is more of an encapsulation. WAV supports Linear PCM plus more than 104 other audio formats, most of which are proprietary coder-decoders (codecs) and many of which use some method of compression. Supporting WAV is tantamount to supporting all the codecs which WAV supports. That is not in line with the basic goals of the encapsulation proposed here.

2.2 Amplitude Compression with No Streaming

Logarithmic PCM includes two algorithms which were proposed in the *G*.711 ITU-T Recommendations of 1988 [6] operating at a sampling rate of 8-kHz with 8-bits per sample (64-kbps) with extensions to 80-kbps and 96-kbps as prescribed by the wide-band extension of *G*.711.1 [10]. In this scenario, the amplitude of the signal goes through some logarithmic transformation to increase the dynamic range of the signal. This conserves the number of bits needed to represent a sample. These two algorithms have been very effective techniques and have been used in telephony applications for many years. In the *G*.711 μ-law (PCMU) and *A*-law (PCMA) coding algorithms, each sample is coded to be represented by 8 bits with an 8-kHz sampling rate which amounts to a bit rate of 64 kbps. These two algorithms are known as PCMU and PCMA, respectively. Most telephony products use either PCMU or PCMA for capturing or recording audio. Supporting these algorithms should cover a wide variety of applications.

2.3 Variable Bit-Rate

These days, the first format that may come to mind is MP3. Unfortunately, MP3 is a proprietary format with many patents attached to it. In contrast, OGG Vorbis is an open-source, variable bit-rate format which, in most cases, performs as well as or better than MP3. Vorbis is the codec and OGG [9, 4] is the encapsulating mean for delivering the Vorbis codec.[2] There are also many open-source tools available including a library called LibAO which is available from the XIPH Open-Source Community for free.[3]

2.4 The Streaming Case

The OGG media stream [9, 4] may be used to stream audio (and video). It is included here as the streaming encapsulation technique. It is completely open-source and can be used with many codecs including MP3. It is, however, recommended that OGG Vorbis be used in conjunction with the OGG media stream to achieve a streaming objective.

3 Header

Table 3 contains the fields of the proposed data header. It (in conjunction with Table 2) constitutes the core of this proposal. After the proposed header, the data format will follow, either as a whole or in the form of a stream which is handled by the OGG header immediately following the proposed header.

In a typical speaker recognition session there may be different *Instances* of audio which may have common information such as the sampling rate, the sample size, the number of channels, etc. This proposal assumes that any such feature will be set once as a default value and that it may be overridden later on, per instance, as the local instance information may change from the overall SIV session information.

ByteOrder is a two-byte, binary code which is written at the time of the creation of the data. It is written as `0xFF00`. When the data is read, if it is read as `0xFF00`, it means that the machine reading the data has the same byte order as the machine writing the data. If it is read as `0x00FF`, it means that the machine reading the data has a different byte order than the machine writing the data and that triggers a byte-swap which is applied to all subsequent information over one-byte in length.

FileLengthInSamples is a convenience measure for using LPCM, PCMU and PCMA. For these cases, *FileLengthInSamples* may be deduced from the *FileLengthInBytes*, *NumberOfChannels*, *SamplingRate* and *BitsPerSample*. It is not, however, readily computable for formats with a variable bit-rate compression. In order for it to be independent of the information which may be embedded in the encapsulated headers of OGG Vorbis, OGG Media Stream or any other format which may be added in the future, this value is included in the proposed header. Since *FileLengthInSamples* is designed for convenience, it may be set to 0.

AudioFullSecondsOf and *AudioRemainderSamples* define *FileLengthInSamples* when the number of samples is so large that an overflow may occur. *AudioFullSecondsOf* is the total number of seconds (in integer form) where the fractional remainder has been truncated. *AudioRemainderSamples* denotes the number of samples remaining in that truncated remainder. For example, if the total audio is 16.5 seconds long and if the sampling rate is 8-kHz, then *AudioFullSecondsOf* will be 16. The truncated remainder will then be 0.5 seconds which multiplied by 8000-Hz will produce 4000 samples which means the value of *AudioRemainderSamples* is 4000. This method of

Type	Variable	Description
U16	ByteOrder	The value is 0xFF00 and it is set by the audio file producer
U16	HeaderSize	Size of the header in bytes
Boolean	Streaming	This will 0 for non-streaming and 1 for streaming. This boolean variable is redundant since the AF_FORMAT for streaming audio is greater than 0x0FFF. However, it is used for convenience.
U64	FileLengthInBytes	In Bytes not including the header
U64	FileLengthInSamples	In Number of samples
U16	AudioFormat	See AF_FORMAT macros
U16	NumberOfChannels	Number of channels, *N.B.*, Channel data alternates
U32	SamplingRate	Sampling rate in samples per second – This is the audio sampling rate and not necessarily the sampling rate of the carrier which may be variable.
U64	AudioFullSecondsOf	It is the truncated number of seconds of audio
U32	AudioRemainderSamples	This is the number of samples of audio in the remainder which was truncated by the above variable
U16	BitsPerSample	Number of bits per sample, may be 0 for formats which use variable bits

Table 3: Audio Format Header

handling of the total number of seconds of audio avoids the use of floating point numbers which are most problematic in cross-platform interchanges. It also supports very long files where specifying the total number of samples can lead to an overflow.

Acronym	Description	Acronym	Description
ANSI	American National Standards Institute	SC	Subcommittee
INCITS	InterNational Committee for Information Technology Standards	SIV	Speaker Identification and Verification
		VBWG	Voice Browser Working Group
ISO	International Organization for Standardization	WG	Workgroup
JTC	Joint ISO/IEC Technical Committee	U8,U16,U32,U64	Unsigned 8, 16, 32 or 64-bit storage

Table 4: Acronyms and Abbreviations

4 Conclusion

The proposal set forth in this paper demonstrates that it is not only possible to standardize audio formats but that it can be achieved through the use of widely-used standard formats. The incorporation of this proposal (or a comparable standards-based formulation) would support the interoperability that is essential for many of the operations to which an SIV module of VoiceXML 3.0 will be put.

References

[1] Beigi, H.: Fundamentals of Speaker Recognition. Springer, New York (2009). ISBN: 978-0-387-77591-3

[2] Community, T.X.O.S.: *0.8* 1.2 Vorbis I Specifications (2004)

[3] Community, T.X.O.S.: LibAO OGG Audio API (2004)

[4] Goncalves, I., Pfeiffer, S., Montgomery, C.: Ogg Media Types. RFC 5334 (Proposed Standard) (2008). URL http://www.ietf.org/rfc/rfc5334.txt

[5] ISO: ISO/JTC1 SC37 WG3, Biometric Data Interchange Format (2009). URL http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=313770

[6] ITU-T: G.711 Pulse Code Modulation (PCM) of Voice Frequencies. ITU-T Recommendation (1988). URL http://www.itu.int/rec/T-REC-G.711-198811-I/en

[7] Markowitz, J.A.: Project 1821 - INCITS 456:200x, Information Technology - Speaker Recognition Format for Raw Data Interchange (SIVR-1) (2009). URL http://www.incits.org/scopes/bsr8_1821.htm

[8] Markowitz, J.A.: Standards for speaker recognition. In: S.Z. Li (ed.) Encyclopedia of Biometrics. Springer, New York (2009). ISBN: 978-0-387-73003-5

[9] Pfeiffer, S.: The Ogg Encapsulation Format Version 0. RFC 3533 (Informational) (2003). URL http://www.ietf.org/rfc/rfc3533.txt

[10] Sollaud, A.: RTP Payload Format for ITU-T Recommendation G.711.1. RFC 5391 (Proposed Standard) (2008). URL http://www.ietf.org/rfc/rfc5391.txt