Contents for Subpart 6

Subpart 6 : TTSI

6.1 Scope

This subpart of ISO/IEC 14496-3 specifies the coded representation of MPEG-4 Audio Text-to-Speech (M-TTS) and its decoder for high quality synthesized speech and for enabling various applications. The exact synthesis method is not a standardization issue partly because there are already various speech synthesis techniques.

This subpart of ISO/IEC 14496-3 is intended for application to M-TTS functionalities such as those for facial animation (FA) and moving picture (MP) interoperability with a coded bitstream. The M-TTS functionalities include a capability of utilizing prosodic information extracted from natural speech. They also include the applications to the speaking device for FA tools and a dubbing device for moving pictures by utilizing lip shape and input text information.

The text-to-speech (TTS) synthesis technology is recently becoming a rather common interface tool and begins to play an important role in various multimedia application areas. For instance, by using TTS synthesis functionality, multimedia contents with narration can be easily composed without recording natural speech sound. Moreover, TTS synthesis with facial animation (FA) / moving picture (MP) functionalities would possibly make the contents much richer. In other words, TTS technology can be used as a speech output device for FA tools and can also be used for MP dubbing with lip shape information. In MPEG-4, common interfaces only for the TTS synthesizer and for FA/MP interoperability are defined. The M-TTS functionalities can be considered as a superset of the conventional TTS framework. This TTS synthesizer can also utilize prosodic information of natural speech in addition to input text and can generate much higher quality synthetic speech. The interface bitstream format is strongly user-friendly: if some parameters of the prosodic information are not available, the missed parameters are generated by utilizing preestablished rules. The functionalities of the M-TTS thus range from conventional TTS synthesis function to natural speech coding and its application areas, i.e., from a simple TTS synthesis function to those for FA and MP.

6.2 Definitions

6.2.1 International Phonetic Alphabet; IPA : The worldwide agreed symbol set to represent various phonemes appearing in human speech.

6.2.2 lip shape pattern : A number that specifies a particular pattern of the preclassified lip shape.

6.2.3 lip synchronization : A functionality that synchronizes speech with corresponding lip shapes.

6.2.4 MPEG-4 Audio Text-to-Speech Decoder : A device that produces synthesized speech by utilizing the M-TTS bitstream while supporting all the M-TTS functionalities such as speech synthesis for FA and MP dubbing.

6.2.5 moving picture dubbing : A functionality that assigns synthetic speech to the corresponding moving picture while utilizing lip shape pattern information for synchronization.

6.2.6 M-TTS sentence : This defines the information such as prosody, gender, and age for only the corresponding sentence to be synthesized.

6.2.7 M-TTS sequence : This defines the control information which affects all M-TTS sentences that follow this M-TTS sequence.

6.2.8 phoneme/bookmark-to-FAP converter : A device that converts phoneme and bookmark information to FAPs.

6.2.9 text-to-speech synthesizer : A device producing synthesized speech according to the input sentence character strings.

6.2.10 trick mode : A set of functions that enables stop, play, forward, and backward operations for users.

6.3 Symbols and abbreviations

F0 fundamental frequency (pitch frequency)

DEMUX demultiplexer

FA facial animation

FAP facial animation parameter

ID identifier

IPA International Phonetic Alphabet

MP moving picture

M-TTS MPEG-4 Audio TTS

STOD story teller on demand

TTS text-to-speech

6.4 MPEG-4 audio text-to-speech bitstream syntax

6.4.1 MPEG-4 audio TTSSpecificConfig

```
TTSSpecificConfig() {
    TTS_Sequence()
}
```


Table 6.4.1 – Syntax of TTS_Sequence

Syntax	No. of bits	Mnemonic
TTS_Sequence() {		
TTS_Sequence_ID	**5**	**uimsbf**
Language_Code	**18**	**uimsbf**
Gender_Enable	**1**	**bslbf**
Age_Enable	**1**	**bslbf**
Speech_Rate_Enable	**1**	**bslbf**
Prosody_Enable	**1**	**bslbf**
Video_Enable	**1**	**bslbf**
Lip_Shape_Enable	**1**	**bslbf**
Trick_Mode_Enable	**1**	**bslbf**
}		

6.4.2 MPEG-4 audio text-to-speech payload

```
AlPduPayload {
    TTS_Sentence()
}
```

Table 6.4.2 – Syntax of TTS_Sequence

Syntax	No. of bits	Mnemonic
TTS_Sentence() {		
TTS_Sentence_ID	**10**	**uimsbf**
Silence	**1**	**bslbf**
if (Silence) {		
Silence_Duration	**12**	**uimsbf**
}		
else {		
if (Gender_Enable) {		
Gender	**1**	**bslbf**
}		
if (Age_Enable) {		
Age	**3**	**uimsbf**
}		
if (!Video_Enable && Speech_Rate_Enable) {		
Speech_Rate	**4**	**uimsbf**
}		
Length_of_Text	**12**	**uimsbf**
for (j=0; j<Length_of_Text; j++) {		
TTS_Text	**8**	**bslbf**
}		
if (Prosody_Enable) {		
Dur_Enable	**1**	**bslbf**
F0_Contour_Enable	**1**	**bslbf**
Energy_Contour_Enable	**1**	**bslbf**
Number_of_Phonemes	**10**	**uimsbf**
Phoneme_Symbols_Length	**13**	**uimsbf**
for (j=0 ; j<Phoneme_Symbols_Length ; j++) {		
Phoneme_Symbols	**8**	**bslbf**
}		
for (j=0 ; j<Number_of_Phonemes ; j++) {		
if(Dur_Enable) {		
Dur_each_Phoneme	**12**	**uimsbf**
}		
if (F0_Contour_Enable) {		
Num_F0	**5**	**uimsbf**
for (k=0; k<Num_F0;k++) {		
F0_Contour_each_Phoneme	**8**	**uimsbf**
F0_Contour_each_Phoneme_Time	**12**	**uimsbf**
}		
}		
if (Energy_Contour_Enable) {		

Energy_Contour_each_Phoneme	**8*3=24**	**uimsbf**
}		
}		
}		
if (Video_Enable) {		
Sentence_Duration	**16**	**uimsbf**
Position_in_Sentence	**16**	**uimsbf**
Offset	**10**	**uimsbf**
}		
if (Lip_Shape_Enable) {		
Number_of_Lip_Shape	**10**	**uimsbf**
for (j=0 ; j<Number_of_Lip_Shape ; j++) {		
Lip_Shape_in_Sentence	**16**	**uimsbf**
Lip_Shape	**8**	**uimsbf**
}		
}		
}		
}		

6.5 MPEG-4 audio text-to-speech bitstream semantics

6.5.1 MPEG-4 audio TTSSpecificConfig

TTS_Sequence_ID This is a five-bit ID to uniquely identify each TTS object appearing in one scene. Each speaker in a scene will have distinct TTS_Sequence_ID.

Language_Code When this is "00" (00110000 00110000 in binary), the IPA is to be sent. In all other languages, this is the ISO 639 Language Code. In addition to this 16 bits, two bits that represent dialects of each language is added at the end (user defined).

Gender_Enable This is a one-bit flag which is set to ‘1’ when the gender information exists.

Age_Enable This is a one-bit flag which is set to ‘1’ when the age information exists.

Speech_Rate_Enable This is a one-bit flag which is set to ‘1’ when the speech rate information exists.

Prosody_Enable This is a one-bit flag which is set to ‘1’ when the prosody information exists.

Video_Enable This is a one-bit flag which is set to ‘1’ when the M-TTS decoder works with MP. In this case, M-TTS should synchronize synthetic speech to MP and accommodate the functionality of ttsForward and ttsBackward. When VideoEnable flag is set, M-TTS decoder uses system clock to select adequate TTS_Sentence frame and fetches Sentence_Duration, Position_in_Sentence, Offset data. TTS synthesizer assigns appropriate duration for each phoneme to meet Sentence_Duration. The starting point of speech in a sentence is decided by Position_in_Sentence. If Position_in_Sentence equals 0 (the starting point is the initial of sentence), TTS uses Offset as a delay time to synchronize synthetic speech to MP.

Lip_Shape_Enable This is a one-bit flag which is set to ‘1’ when the coded input bitstream has lip shape information. With lip shape information, M-TTS request FA tool to change lip shape according to timing information (Lip_Shape_in_Sentence) and predifined lip shape pattern.

Trick_Mode_Enable This is a one-bit flag which is set to ‘1’ when the coded input bitstream permits trick mode functions such as stop, play, forward, and backward.

6.5.2 MPEG-4 audio text-to-speech payload

TTS_Sentence_ID This is a ten-bit ID to uniquely identify a sentence in the M-TTS text data sequence for indexing purpose. The first five bits equal to the TTS_Sequence_ID of the speaker defined in subclause 6.1, and the rest five bits are the sequential sentence number of each TTS object.

Silence This is a one-bit flag which is set to '1' when the current position is silence.

Silence_Duration This defines the time duration of the current silence segment in milliseconds. It has a value from 1 to 4095. The value '0' is prohibited.

Gender This is a one-bit which is set to '1' if the gender of the synthetic speech producer is male and '0', if female.

Age This represents the age of the speaker for synthetic speech. The meaning of age is defined in Table 6.5.1.

Table 6.5.1 – Age mapping table

Age	age of the speaker
000	below 6
001	6 - 12
010	13 - 18
011	19 - 25
100	26 - 34
101	35 - 45
110	45 - 60
111	over 60

Speech_Rate This defines the synthetic speech rate in 16 levels. The level 8 corresponds the normal speed of the speaker defined in the current speech synthesizer, the level 0 corresponds to the slowest speed of the speech synthesizer, and the level 15 corresponds to the fastest speed of the speech synthesizer.

Length_of_Text This identifies the length of the TTS_Text data in bytes.

TTS_Text This is a character string containing the input text. The text bracketed by < and > contains bookmarks. If the text bracketed by < and > starts with FAP, the bookmark is handed to the face animation through the TtsFAPInterface as a string of characters. Otherwise, the text of the bookmark is ignored. The syntax of the bookmarks is defined in ISO/IEC 14496-2.

Dur_Enable This is a one-bit flag which is set to '1' when the duration information for each phoneme exists.

F0_Contour_Enable This is a one-bit flag which is set to '1' when the pitch contour information for each phoneme exists.

Energy_Contour_Enable This is a one-bit flag which is set to '1' when the energy contour information for each phoneme exists.

Number_of_Phonemes This defines the number of phonemes needed for speech synthesis of the input text.

Phonemes_Symbols_Length This identifies the length of Phonemes_Symbols (IPA code) data in bytes since the IPA code has optional modifiers and dialect codes.

Phoneme_Symbols This defines the indexing number for the current phoneme by using the Unicode 2.0 numbering system. Each phoneme symbol is represented as a number for the corresponding IPA. Three two-byte numbers is used for each IPA representation including a two-byte integer for the character, and an optional two-byte integer for the spacing modifier, and another optional two-byte integer for the diacritical mark.

Dur_each_Phoneme This defines the duration of each phoneme in msec.

Num_F0 This defines the number of F0 values specified for the current phoneme.

F0_Contour_each_Phoneme This defines half of the F0 value in Hz at time instant F0_Contour_each_Phoneme_Time.

F0_Contour_each_Phoneme_Time This defines the integer number of the time in ms for the position of the F0_Contour_each_Phoneme.

Energy_Contour_each_Phoneme These 3 8-bit data correspond to the energy values at the start, the middle, and the end positions of the phoneme. The energy value X is calculated as

$$X = \text{int}(50 \log_{10} A_{p-p}),$$

where A_{p-p} is the peak-to-peak value of the speech waveform at the defined position.

Sentence_Duration This defines the duration of the sentence in msec.

Position_in_Sentence This defines the position of the current stop in a sentence as an elapsed time in msec.

Offset This defines the duration of a very short pause before the start of synthesized speech output in msec.

Number_of_Lip_Shape This defines the number of lip-shape patterns to be processed.

Lip_Shape_in_Sentence This defines the position of each lip shape from the beginning of the sentence in msec.

Lip_Shape This defines the indexing number for the current lip-shape pattern to be processed that is defined in Table 12.5 in ISO/IEC 14496-2.

6.6 MPEG-4 audio text-to-speech decoding process

The architecture of the M-TTS decoder is described below and only the interfaces relevant to the M-TTS decoder are the subjects of standardization. The number above each arrow indicates the section describing each interface.

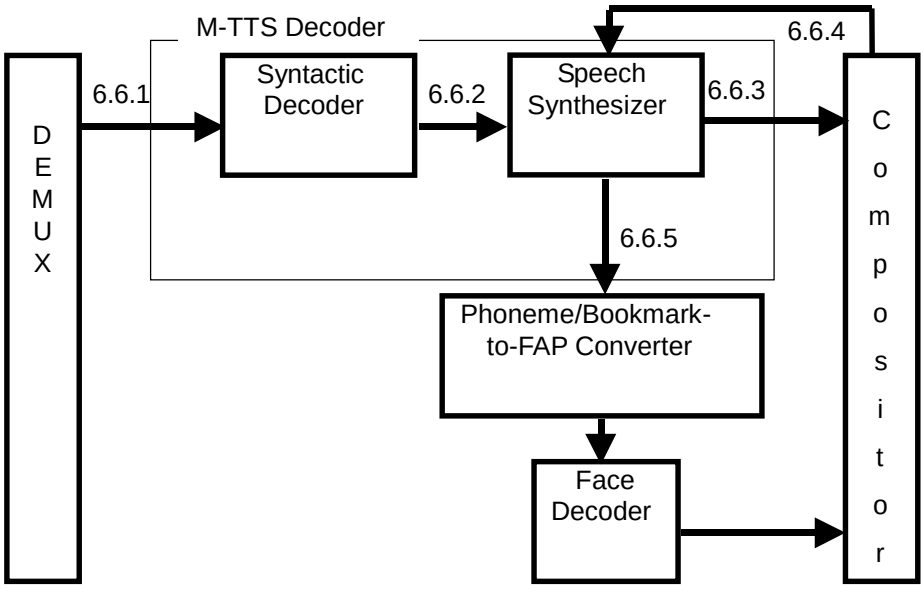

Figure 6.6.1 - MPEG-4 Audio TTS decoder architecture

In this architecture the following types of interfaces is distinguished:

Interface between DEMUX and the syntactic decoder

Interface between the syntactic decoder and the speech synthesizer

Interface from the speech synthesizer to the compositor

Interface from the compositor to the speech synthesizer

Interface between the speech synthesizer and the phoneme/bookmark-to-FAP converter

6.6.1 Interface between DEMUX and syntactic decoder

Receiving a bitstream, DEMUX passes coded M-TTS bitstreams to the syntactic decoder.

6.6.2 Interface between syntactic decoder and speech synthesizer

Receiving a coded M-TTS bitstream, the syntactic decoder passes some of the following bitstreams to the speech synthesizer.

Input type of the M-TTS data: specifies synchronized operation with FA or MP

Control commands stream: Control command sequence

Input text: character string(s) for the text to be synthesized

Auxiliary information: Prosodic parameters including phoneme symbols

Lip shape patterns

Information for trick mode operation

The pseudo-C code representation of this interface is defined in subclause 6.4.

6.6.3 Interface from speech synthesizer to compositor

This interface is identical to the interface for digitized natural speech to the compositor. The dynamic range is from – 32767 to + 32768.

6.6.4 Interface from compositor to speech synthesizer

This interface is defined to allow the local control of the synthesized speech by users. This user interface supports trick mode of the synthesized speech in synchronization with MP and changes some prosodic properties of the synthesized speech by using the ttsControl defined as follows:

Table 6.6.1 – Syntax of ttsControl()

Syntax	No. of bits	Mnemonic
ttsControl() {		
ttsPlay()		
ttsForward()		
ttsBackward()		
ttsStopSyllable()		
ttsStopWord()		
ttsStopPhrase()		
TtsChangeSpeechRate()		
TtsChangePitchDynamicRange()		
TtsChangePitchHeight()		
TtsChangeGender()		
ttsChangeAge()		
}		

The member function ttsPlay allows a user to start speech synthesis in the forward direction while ttsForward and ttsBackword enable the user to change the starting play position in forward and backward direction, respectively. The ttsStopSyllable, ttsStopWord, and ttsStopPhrase functions define the interface for users to stop speech synthesis at the specified boundary such as syllable, word, and phrase. The member function ttsChangeSpeechRate is an interface to change the synthesized speech rate. The argument speed has the numbers from 1 to 16. The member function ttsChangePitchDynamicRange is an interface to change the dynamic range of the pitch of synthesized speech. By using the argument of this function, level, a user can change the dynamic range from 1 to 16. Also a user can change the pitch height from 1 to 16 by using the argument height in the member function ttsChangePitchHeight. The member functions ttsChangeGender and ttsChangeAge allow a user to change the gender and the age of the synthetic speech producer by assigning numbers, as defined in subclause 6.5.2, to their arguments, gender and age, respectively.

6.6.5 Interface between speech synthesizer and phoneme/bookmark-to-FAP converter

In the MPEG-4 framework, the speech synthesizer and the face animation are driven synchronously. The speech synthesizer generates synthetic speech. At the same time, TTS gives phonemeSymbol and phonemeDuration as well as bookmarks to the Phoneme/Bookmark-to-FAP converter. The Phoneme/Bookmark to FAP converter generates relevant facial animation according to the phonemeSymbol, the phonemeDuration and bookmarks. Further description of the Phoneme/Bookmark to FAP converter is provided in ISO/IEC 14496-2.

The synthesized speech and facial animation have relative synchronization except the absolute composition time. The synchronization of the absolute composition time comes from the same composition time stamp of the TTS bitstream. If the Lip_Shape_Enable is set, the Lip_Shape_in_Sentence is used to generate the phonemeDuration. Otherwise, the TTS provides phoneme durations. The speech synthesizer generates stress and/or wordBegin bits when the corresponding phoneme has stress and/or start of a word, respectively.

Within the MTTS_Text, the beginning of a bookmark for using facial animation parameters is identified by '<FAP'. The bookmark lasts until the closing bracket '>'

A bookmark is handed to the TtsFAPInterface with the phoneme of the next word of the current sentence following the bookmark. If there is no word after the bookmark, the bookmark is handed to the TtsFAPInterface with the last phoneme of the previous word in the current sentence. In order to allow animation of complex expressions and motion, a sequence of up to 40 bookmarks is allowed without words between them. The starttime defines the time in msec relative to the beginning of the M-TTS sequence when the phoneme will start playing.

The class ttsFAPInterface defines the data structure for the interface between the speech synthesizer and the phoneme-to-FAP converter.

Table 6.6.2 – Syntax of TtsFAPInterface()

Syntax	No. of bits	Mnemonic
TtsFAPInterface() {		
PhonemeSymbol	8	uimsbf
PhonemeDuration	12	uimsbf
f0Average	8	uimsbf
Stress	1	bslbf
WordBegin	1	bslbf
Bookmark		char *
Starttime		long int
}		