音频媒体控制语言 AMCL

Audio Media ControL

胡其炜，蔡莲红
（清华大学 计算机科学与技术系，北京 100084）

摘要：AMCL 是采用了 VRML 对象化结构设计的音频媒体控制语言。它参照目前各种语音标注语言，为文语转换系统(TTS)提供了一套标准的、有效的、简洁的韵律控制符号；同时，加入了对于音响效果的控制和对多个音源的位置与时序的定义；因而适合于描述多媒体制作、和 Internet 网络传输中的复杂音响效果。

关键词：语音合成，TTS，韵律标注，媒体控制

引言

为语音合成系统设计一套用于韵律控制的标注语言是必需的。在国际上，一些组织针对语音合成的需要，在通用标注语言 SGML 的基础上先后提出了 SSML，JSML，STML 等标注语言。但是，由于自然语言本身的复杂性和特殊性，要定义一种通用的标注形式对各种自然语言进行统一的、精确的韵律控制，是一个极端困难的工作。正因为如此，目前的各种语音标注语言的给出的功能框架都是很粗略的。我们依据已有的标注语言，研究了适用于汉语韵律描述的标注方式，并且进一步扩展得到一种通用的音频媒体控制语言 AMCL(Audio Media Control Language)。AMCL 在描述复杂语音韵律模式之外，加入了对于音响效果的控制和对多个音源的位置与时序的定义，因此适合于描述和控制目前计算机应用中更具普遍意义的一般的音频媒体。

语言概述

AMCL 试图为描述由计算机合成的复杂的声音效果提供一种标准方法；其主要功能有：描述复杂语音模型和韵律规则；组合并播放合成语音(Speech)、合成乐音(MIDI)与波形数据(Wave)；定义空间中多个音源的位置、运动方式、播放时序；定义常用的数字化音频操作。用户可以通过一个或多个 AMCL 文档对需要制作的音效进行置标描述，在 AMCL 播放器中播放出来。

对 AMCL 语言的开发主要为了解决下列问题：

1、为文语转换系统(TTS)定义一套标准的、有效的、简洁的韵律控制符号。一方面，它方便了使用者对 TTS 韵律合成的人工干预；另一方面，它为 TTS 系统的开发者提供了通用的控制接口，和有力的测试工具；
2、为描述多媒体制作中的复杂音响效果定义一套原语；
3、提供在广域网络（如 Internet）环境下的高质量混合音频媒体的传送方案。

从 Internet 的角度看，AMCL 文档可以作为一种新的 MIME 类型嵌入到 HTML 文档中，而把 AMCL 的播放器作为 WWW 浏览器平台的一个功能模块(如 Netscape 的 Plug-in)。这样，

AMCL 文本就可以利用现有的 HTTP 协议在 WWW 上传送，并在客户端播放出预先制作的特殊音效。

AMCL 语言的设计应当达到以下的要求：

- **平台无关性**
  AMCL 的定义独立于硬件平台、发音设备、操作系统、TTS 引擎和网络设施。
- **简单性**
  简单性包含两个方面：从人的角度来说，它是易于理解和使用的；从计算机角度来说，它是易于分析和实现的。
- **一致性**
  无论是通过计算机分析或是人的理解，AMCL 及其扩展均不能导致二义性的描述出现。
- **全面性**
  AMCL 必须足够描述所有常用的和必需的语音特性及音频模式。
- **可扩展性**
  AMCL 的定义留有适当的接口，可以针对特定的运行环境做出必要的功能性扩充；同时，该扩充不影响在其它环境下的 AMCL 播放器的正常运行。
- **在低带宽网络上的良好工作能力**
  为了在低带宽网络上较好地工作，AMCL 应当保证较低的数据率，和在一定时间段内声音播放的连贯性。

AMCL 的定义分为两个子系统：语音韵律置标和媒体控制语言。

语音韵律置标

韵律置标主要是为语音合成设计的。在 AMCL 的语音韵律置标层，我们定义了一种新的韵律置标系统，称为简单韵律置标符号（Simple Text Tagging Symbol，STTS 或 TTS[2]）。简单文本置标符号是基本 TTS 系统功能的一个有效的补充，通过文本内嵌的置标符号来控制合成的韵律效果。和普通的文语转换相比，内嵌的置标容易通过人工干预的方式提高合成的表现力和自然度。简单韵律置标符号是一套独立的置标命令，具有以下特点：

1、 短小、简单。由于文语转换以汉字文本为主体，置标符号在结构上和篇幅上都应以不影响人对原始文本的理解为限。
2、 易于理解和使用。
3、 主要用于控制 TTS 的文本分析过程(间接控制方式)或说明字、词的韵律参数(直接控制方式)。
4、 置标符号集可以扩充。置标的扩充仍应遵循简单、一致、无歧义的原则。

韵律置标是文本中包括在一对反斜杠之间的部分。除了少量的特殊置标外，置标的通常格式为：

\C\ 或 \C n\

其中，C 表示命令，由一个或多个英文大写字母组成；n 表示参数。n 可以包括除了反斜杠和其它不可见 ASCII 字符外的所有字符。在一个韵律置标内出现的回车、换行、空格和制表字符都将被视作参数的语法分隔符。

置标的使用遵循以下规则：

- 两个连续的反斜杠表示一个独立的(\)字符，不作为置标的起始；
- 置标在遇到一个反斜杠后即结束；
- 置标的作用范围根据其意义而不同。所有置标的作用域应紧接在置标出现的位置之后；
- 置标可以在文本中任意位置出现，但文本的解释器不保证任何出现的置标都会得到处理；
- 如果几个置标的作用域重叠，它们在其作用域重叠部分对合成的作用是叠加的。

简单韵律置标符号直接控制合成的韵律特征，而与合成器本身的合成算法、实现细节无关。标记根据其功能分为如下的几类：

一、通用置标（GENERAL TAGS）

注释

注释字符串\　注释由一对反斜杠及其内部的星号包括起来。

文本编码说明

\CODE [n]\　n 代表处理文本的编码。预定义的汉字内码有：

GB：国家标准码；BIG5：Big5 码；GBK：国家标准扩展码

如果文本分析器遇到非法的汉字内码，则认为是 ASCII 码。

如果 n 被省略，则认为是 GBK 码。合成引擎对编码的扩充可以参考国际标准的的 ISO 语言代码(参见 ISO639 标准)。

e.g.　\CODE=BIG5\ 指定被分析文本为 Big5 码。

系统复位

\RST\　合成系统的所有参数回复到缺省值。

提示标记

\NOTE [note]\　在文本的当前位置放置一个事件标记。合成引擎可以忽略此标记，也可以在播放过程中遇到此标记时对程序产生一个通知。在通知的内容中给出 note 字符串。

设置嗓音

\VOC [character]\　character 表示嗓音代码，预定义的嗓音代码有：

gm：一般男声；gf：一般女声；gc：一般童声

缺省值为 gm。即\VOC\ → \VOC gm\。合成引擎可以自行扩充嗓音代码。更换音库可以同时更换用于模拟相应说话人的韵律规则。

二、文本分析控制

篇章类型

\CTX [context]\　context 指定该置标之后的文本类型。合成引擎可以据此调整文本分析方式和韵律模型。预定义的 context 值有：

default：缺省类型；

novel：文学作品；poem：诗；technical：技术性文档；program：程序文档；maths：数学表达式；email：电子邮件；address：普通邮件地址；time：日期或时间；literal：逐字读出；

缺省值为 default。

文本格式

\CR [1|0]\　控制其后的回车/换行字符的作用。当设置为 1 时，回车/换行字符被用作强制分词符，并进行一个短暂的停顿；否则文本分析器忽略文本中的回车/换行字符。

缺省设置为 0。

\SP [1|0]\　控制其后的空格和制表符的作用。当为 1 时，空格和制表符被用作强制分词符，并进行一个短暂的停顿；否则文本分析器忽略文本中的空格和制表符。

缺省设置为 0。

数字读法

\DG [tele|num|def]\　n 表示数字读出方式：tele 表示按电话码无单位方式读出；num 表示按普通数字读出。def 表示按照合成系统缺省的分析结果读出。

缺省值为 0。

e.g.　\DG 1\1997 年\DG\7 月 25 日

三、注音

字符注音

\PY=[pinyin]\　pinyin 表示带音调汉语拼音字符串。若音调省略，则认为是轻声。拼音使用小写字母。如果在此标记中只列出一个拼音，则表示标明其后的一个汉字的读音。后续的一个汉字参加分词，按照注出的拼音发音；如果一次列出多个拼音，则这些拼音按词语发音，而不影响后续的汉字的发音。多个拼音之间用空格隔开。

e.g.　\PY zeng1\曾老师来了没有？

四声变调：\PY a1 a2 a3 a4\ 和一个轻声。

音标

\IPA [ipa]\　ipa 表示一个国际音标串。合成引擎按此音标串发音。

读音

\PHN [string]\　string 代表一个由目前定义的语言编码的字符串，定义其后一个字或符号的发音。合成引擎按此字符串发音。

e.g.　总价值为 1450\PHN 美元\$。

特殊音效

\FX [effects]\　effects 代表合成系统内置的特殊音效片段的名称。合成系统可以自行确定特殊音效的播放是否受到韵律置标的控制，而修改具体的播放效果。

四、韵律置标

局部重音

\EM [n1][n2]\　n 表示此置标后一个汉字的重音级别。正常汉字为 n = 5。其余值定义为：

9 ： 最重音；8 ： 重音；7 ： 较重音；6 ： 微重音；4 ： 微轻音；3 ： 较轻音；2 ： 轻音；1 ： 极轻音；0 ： 最轻音。

一个 n 值表示字的整体种音级别。2 个 n 值表示字内的重音变化。缺省为 n1=5，n2 为空。合成系统可以自行定义重音的具体实现及一个字的重音对附近字的影响。

局部停顿

\P[n]\　n 表示停顿时间。n 值最小为 0；9 为正常的句间停顿时间。n 值增加，代表的停顿时间不会缩短。即 n1 > n2 时，有 t1 ≥ t2。具体停顿时间 t=f(n)由合成引擎具体定义。缺省值为 9。

字间停顿

\G[n]\　n 表示在此置标之后的字间停顿，也即字的连读程度。最小值为 0。n = 5 是正常停顿时间。规定 n1 > n2 时，停顿时间 t1 ≥ t2。其中 t=f(n)由合成系统具体定义。缺省值为 5。

如果在 n 数字前有一个加号(+)，表示停顿基准提高相应的级别；一个减号(-)表示停顿的基准降低相应的级别。此后的停顿时间按新的基准进行调整。

e.g.　　请你\G+5\一字一顿\G-5\地说话。

速度

\S[n]\　n 表示在此置标之后的整体朗读速度。最小值为 0。n = 5 是正常朗读速度。规定 n1 > n2 时，单位时间读出的字数 w1 ≤ w2。其中 w=f(n)由合成系统具体定义。缺省值为 5。

如果在 n 数字前有一个加号(+)，表示速度基准提高相应的级别；一个减号(-)表示速度的基准降低相应的级别。此后的朗读速度按新的基准进行调整。

e.g.　　请你\S+5\一目十行\S-5\地说话?

音量

\V[n]\　n 表示此置标后的整体音量。最小值为 0，规定 9 是系统的最大音量。n1 > n2 时，音量 v1 ≥ v2。其中 v=f(n)由合成系统具体定义。缺省值为 5。

如果在 n 数字前有一个加号(+)，表示音量基准提高相应的级别；一个减号(-)表示音量基准降低相应的级别。此后的播放音量按新的基准进行调整。

基频

\I[n][,range]\　n 表示此置标后的整体基频。最小值为 0，n = 5 是正常朗读的基频。规定 n1 > n2 时，基频 p1 ≥ p2。其中 p=f(n)由合成系统具体定义。缺省值为 5。

如果在 n 数字前有一个加号(+)，表示基频基准提高相应的级别；一个减号(-)表示基频基准降低相应的级别。此后的播放基频按新的基准进行调整。

range 表示调域的范围。最小值为 0，n = 5 是正常朗读的范围。规定 n1 > n2 时，范围 r1 ≥ r2。其中 r=f(n)由合成系统具体定义。缺省值为 5。range 值前添加加减号的规则同上。省略 range 表示不修改当前的调域。

四、特殊韵律控制

朗读风格

\CHR [style]\　指定需要模拟的朗读情绪。预定义的情绪如下：

normal：平淡；angry：生气；happy：快乐；sad：悲伤；kidding：戏谑；serious：严肃；fear：胆怯；sarcastic：嘲讽

特殊嗓音

\VOX [voice]\　指定需要合成的特殊嗓音：预定义的特殊嗓音有：

normal：正常；fake：假嗓；gnash：切齿音；sign：叹气音；broken：断续音；tremble：颤抖音；whisper：耳语

音乐置标

\M [m]\　m 代表一音乐置标串。音乐置标串由若干音乐置标符的有序排列组成，置标符之间没有空白字符，回车或其它不可见 ASCII 字符。音乐置标控制紧接其后的一个字或发音标记的韵律。

定义的音乐置标符有：

On	n=0..6，设置新的 8 音程。缺省为 3
<	在目前音程基础上降低一个音程
>	在目前音程基础上升高一个音程
Rn	n 为一个音符，表示音 A=n。
Pn	n=1..64，产生一个休止符。n=1 是全音符长度。64 是 1/64 音符长度。
Vn	n=0..9，表示音量。0 最小，9 音量最大。
Tn	n=1..256，表示每分钟播放的 1/4 音符数。该置标说明了放音节奏。
-	放在置标串末尾，表示合成引擎在该字末尾应考虑与下一字的连续发音。
N(n)	产生一个事件 n，n 为字符串。

音符格式为：

C[s][n]　C 为音符，由小写 c, d, e, f, g, a, b 表示。

S 可以省略，或为下列符号之一，表示音的升降：

：升高半音。

！：降低半音。

缺省 s=空。

n=1..64，1 是全音符长度，64 为 1/64 音符长度，缺省为 n=1。

合成系统应按顺序解释音乐置标，并修改汉字的声学参数。

媒体控制语言

媒体控制部分采用的是类似于 VRML（版本 1.0）的对象化描述方式。

元指令

注释

一个 AMCL 文档中，在一个井字(#)符号之后直到行尾的部分是注释。播放器一般可以

忽略注释的内容。由以下字符串开始的是 AMCL 的元指令：

#AUTHOR [name]	定义文档的作者
#KEYWORD [keywords]	定义文档的关键字以备搜索
#TITLE [title]	定义文档的主题
#INCLUDE [url]	包含另一个 AMCL 文档

音源对象定义

对象由对象名，对象类型及相应的属性组成。一般的对象定义语句为：

```
DEF 对象名 对象类型
{
    属性 1   属性值
    属性 2   属性值
    ……
}
```

不同的对象类型具有不同的属性。如果一个对象在定义时没有设置某个属性，则设置为播放系统定义的缺省值。在定义属性值时也可以使用以下的子句：

USE 子句

格式：USE [对象名]

拷贝另一个对象的同名属性。被 USE 的对象必须具有相同的属性。否则该子句被忽略。

AS 子句

格式：AS [对象名]

引用另一个对象的同名属性。如果被引用的对象的属性发生变化，则本对象的同名属性作相同的改变。

URL 子句

格式：URL [url 名称]

表示属性的内容由一个 URL 文档指定。URL 可以是一个由[RFC1738]指定的全局定位器，或是一个由[RFC 1808]指定的局部定位器。

对象定义

对象分为三类：音源、监听器、滤波器。

音源定义

音源对象具有以下的公共属性：

Volume [volume]	音量（百分数，最大为 100%）
Loop [on|off]	是否循环（缺省为 off）
Pos {x y z}	位置，x, y, z 采用浮点表示，左手坐标系
Orient {x y z}	音源的朝向，x, y, z 采用浮点表示，左手坐标系
Ambient [x]	音源的环境区，x 为浮点数
Attenuation [x]	音源的衰减率，x 为浮点数
Filters {filter list}	音源的滤波器列表。表中为滤波器的名字，多个名字之间用空格隔

开。滤波器对象也可以用于监听器，此时成为一个全局的滤波器。

以下的说明中只列出与音源类型有关的特殊属性。

合成语音

```
DEF [objname] TEXT
{
    Speaker [SpeakerName]      # 说话人
    Text {Text content}        # 文本内容；可以带有 STTS 置标
}
```

合成乐音

```
DEF [objname] MUSIC
{
    Instrument [ins]           # 乐器名称，或 General MIDI 中的编号
    Note {notes}               # 标记单个乐器通道的乐谱串，使用 STTS 中的音乐置标符
}
```

波形

```
DEF [objname] WAVE
{
    Data [name]                # 音效名称；波形的编码方式由播放器自动检测。播放器可以有选择地支持常见的一些编码格式，如 ADPCM，True Speech，Real Audio，但必须支持 PCM 编码。
}
```

监听器

```
DEF [objname] LISTENER
{
    Pos {x y z}                # 位置
    Orient {x y z}             # 朝向
    Filter {filter list}       # 滤波器
    Volume [n]                 # 音量
}
```

缺省监听器的名字为 default，位置为{0 0 0}，朝向{0 0 1}，滤波器为空

滤波器

```
DEF [objname] FILTER
{
    FREQRESPONSE               # 频率过滤器
    {
        hertz   amplify        # 频率-放大倍数对，描述频率响应曲线；使用线性插值或样条曲线插值。
    }
    ENVELOPE                   # 包络
```

```
    {
        time    amplify             # 时间-放大倍数对，time 取值[0.0, 1.0]
    }
    PITCHCURVE
    {
        hertz    newhertz           # 频率曲线，由(hertz, newhertz)控制
    }
    TIMECURVE
    {
        time    newtime     # 时间曲线由(time, newtime)控制，为[0.0, 1.0]之间的浮点数。
    }
    PITCHSHIFT [n]          # n 为偏移倍数
    TIMESTRETCH [n]         # n 为时延倍数
}
```

运动定义

属性设置

```
SET [objname] [n]           # n 为浮点数，表示设置动作的时间（秒），在 n 秒后对象的属
                            性达到设置值，过渡的数值可以通过插值得到。n = 0 表示立即
                            设置。
{
    属性        属性值      # 设置相应的属性为新的设置值
    ……
}

LISTEN [listener name]      # 指定当前的监听器
```

时序与事件

```
START [objname]             # 开始播放音源
STOP [objname]              # 停止播放音源
WAIT [objname] [event]      # 等待对象的某个事件。预定义的事件有：start, stop, pause,
                            resume；可以接受文本置标或音乐置标种自定义的事件。
WAIT [n]                    # n 为一浮点数，表示等待的秒数
```

结束语