嵌入式富媒体逻辑控制的标签化脚本

张骥先，罗 蕾

(1.电子科技大学计算机科学与工程学院 成都 610054)

【摘要】通过研究富媒体场景的交互性需求，提出了一种适合嵌入式设备的富媒体场景控制脚本。该脚本采用XML标记语言描述语法，通过定义标签以及属性来表示语义，并且利用场景COM树的遍历流程进行执行，解决了场景的逻辑控制、状态恢复和终端设备能力调用的问题。该脚本可以与场景共用很多相同的模块，具有执行效率高以及消耗资源少的特点。通过与其他脚本进行的测试比较，证实了该脚本具有良好的效果。

关 键 词 交互性； 逻辑控制； 标签语言； 多媒体服务

中图分类号 TP311 **文献标识码** A doi:10.3969/j.issn.1001-0548.2009.06.025

Embedded Rich Media Logic Control Base on Markup Language Script

ZHANG Ji-xian and LUO Lei

(1. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract According to the interactive requirements of rich media scene, a script for control of rich media scene in embedded environment. In this script, XML markup language is adopted to describe grammar. The semantics is represented by defining elements and attributes. The script can be executed by traversing the scene DOM tree and it resolves logic controlling, state recovering, and terminal device capabilities calling of the scene. The script can share lots of same modules with the scene, such as parsing modules and executing modules. Therefore, the script has the characteristics of high execution efficiency and low consumption of resources compared with some other existing scripts.

Key words interactive; logic control; markup language script; multimedia services

富媒体有别于传统媒体，它将文本、图形、图片、动画、音视频等多种媒体对象在时间/空间上进行有机结合，提供丰富的表现形式和交互能力。其主要特征是可基于时间或用户交互的基础上产生动态行为。

富媒体应用与传统的多媒体应用不同，传统的媒体应用多是单一的视频或者音频播放，不能让用户与媒体对象产生互动，缺乏交互性。同时，富媒体应用与传统的网页应用也不同，除了用户点击产生变化外，富媒体应用中内容的改变很大一部分还依赖于时间的推进。富媒体应用在嵌入式设备上的适用范围很广，包括交互式电视、富媒体广告、电子杂志、动态界面、交互式服务(投票、订阅、缴费)等方面，但与此同时，嵌入式设备的处理能力也成为了富媒体应用发展的障碍。

目前嵌入式领域的富媒体应用倾向于使用专门的富媒体引擎，通过定义基于XML规范的轻量级标记语言、脚本及其他技术等展现各种的富媒体应用，如Adobe公司的Flashlite技术、W3C的SVG Tiny 1.2[1]标准以及基于SVG Tiny 1.2标准上的扩展技术，如MPEG组织的LASeR[2]标准、3GPP组织的DIMS[3]标准、NOKIA倡导的MORE[4]方案等。这一类技术的优点是富媒体应用开发周期短，开发难度低，应用制作人员只需简单/无需编程能力，且表现能力丰富，适用范围广。

通过扩展SVG Tiny 1.2标准的富媒体技术[2-3]在响应应用快速变化、媒体展现力强大、应用开发简单等多方面优势明显，成为嵌入式富媒体技术[2-4]的发展趋势，它认为一个富媒体应用包含一个或多个富媒体场景。富媒体场景描述了媒体对象在时间和空间上的布局信息以及交互行为，一个富媒体场景由一个初始场景(initial scene)以及可能有的一系

收稿日期：2008－06－11；修回日期：2009－05－06
基金项目：国家电子发展基金(H04010601W0608138)
作者简介：张骥先(1980－)，男，博士生，主要从事富媒体和下一代移动网络应用相关技术方面的研究.

列含有序号的更新(update)组成，这些更新可以是按时序出现，也可能是通过用户交互产生，可以向原有场景添加、删除、替换媒体对象，使场景内容发生改变，呈现出动态的效果。这种技术可使用SVG Tiny 1.2语言描述富媒体场景。

1 SVG Tiny 1.2的交互性

SVG Tiny 1.2(scalable vector graphics tiny)是W3C组织定义的一种基于XML的、开放标准的、针对嵌入式设备的、最新的二维矢量图形描述语言。由于它是基于XML的，几乎具有XML的所有优点，例如：方便描述及扩展、易于阅读等；同时，它具有支持用户与媒体对象交互、为媒体对象定义时间和空间上的布局，可以描述矢量动画效果，可随意放大缩小图形的优点。但是SVG Tiny 1.2起初并不是用于描述富媒体场景的，对富媒体应用来说，SVG Tiny 1.2在交互性上显得弱了一些。虽然可以加入脚本增强交互性，但同时也增加了终端的负担。

文献[4]所述的富媒体应用中，交互一般分为：

(1) 场景间交互(remote interactive)：场景间交互指的是发生场景切换或者场景更新，一般会涉及到与服务端的交互行为。

(2) 场景内交互(local interactive)：场景内交互是指在当前场景中如何加入更多的操作逻辑和动态元素，使得当前场景的呈现更加生动，从而给用户带来视觉和操作上的改善，一般只涉及到终端设备。

对于场景间交互，目前SVG Tiny 1.2语言可以使用脚本或<a>标签来实现场景间的切换。

对于场景内交互，SVG Tiny 1.2提供了简单的动画控制、事件机制、文字输入以及焦点移动，这些特性都体现了SVG Tiny 1.2的交互性。但仅使用这些特性，依然不能满足富媒体应用的多样性。例如，在富媒体应用中经常需要用到逻辑控制功能来实现不同的效果，比如根据多个条件判断后选择执行不同的动画效果、依次循环执行不同的动画效果、焦点的重新设置，在传统的技术中都是借助于脚本语言实现的。

2 分析与实现

2.1 SVG Tiny 1.2交互性缺陷

SVG Tiny 1.2不具有编程语言的逻辑控制功能，在以往使用轻量级标记语言富媒体技术中，富媒体场景的逻辑控制是采用脚本或者其他编程语言实现的，它们要么必须解释执行，要么之前必须通过编译。这样无疑增加了终端处理的负担，也增加了应用的开发难度。在一些资源受限的平台上，加入脚本引擎实现逻辑变得很困难，不管从执行效率和程序大小方面都难以忍受。另一方面，富媒体应用与传统的网络应用不同，富媒体应用是一种针对媒体和交互的应用，在很多情况下简单的逻辑交互功能结合SVG本身的动画行为就可满足大部分富媒体应用的需求。

参考以往各种web应用或富媒体应用，脚本的功能主要体现在以下几个方面：

(1) 逻辑控制。主要是分支判断，循环和跳转。

(2) 变量的使用。主要用于保存状态以及辅助逻辑的正确执行。

(3) API的调用。如DOM接口的使用、通信功能以及对本地扩展方法的调用。

SVG Tiny 1.2定义了uDOM接口供脚本使用，这类接口在类似GIS的应用中很有用，但是对于富媒体应用，静态的场景居多，需要用到动态创建节点的场合极少，而且在LASeR和DIMS中采用了场景命令的方式也可以达到动态更改场景内容的目的，且更加高效。LASeR和DIMS定义了不同的Profile来适应不同的平台，在BASE Profile上并不推荐使用脚本，LASeR和DIMS为此扩展了部分标签实现来脚本的部分功能，例如<conditional>标签配合场景命令<insert>，<delete>，<repleace>，<active>，<deactive>的使用可以达到uDOM的部分功能；可以关闭或激活场景中不需要执行的动画节点；可以根据场景时间去打开一个新的链接，完成了部分脚本才能完成的功能。但LASeR和DIMS也没有实现逻辑控制的功能，即与SVG一样，只能根据产生的事件去执行定义好的操作，缺乏灵活性。

下面列出了一些富媒体应用中常见的例子，采用上述的技术在不使用脚本的情况下却无法实现。

(1) SVG Tiny 1.2无法使用同一个事件根据条件的不同执行不同的动画，这种有用的功能可以做出类似苹果公司播放器的coverflow的效果或者旋转菜单的效果。图1的例子描述了通过点击红色按键可以循环触发执行多段动画从而改变A对象的位置。原因是富媒体场景无法创建和保存变量，无法执行判断功能。而对于脚本或编程语言来说，这是很简单的，脚本创建一个变量，当事件触发脚本每次执行的时候，脚本根据变量值的不同执行不同的动画。

(2) 针对嵌入式设备设计的富媒体应用，考虑到屏幕大小的因素，很多采用了类似标签页的设计方

式，以达到操作简洁且容纳内容多的效果。但在目前大多数嵌入式富媒体引擎或wap浏览器中，焦点只能按照当前的布局信息或者DOM树的顺序进行移动，在一些节点很多的场景中，在不支持触摸屏的终端上进行信息浏览很不方便。例如，某个富媒体场景中有多个tab页，每个tab页面内有一个封闭的焦点移动序列，在某个tab页内，焦点只能在本页内的对象之间移动，无法移动到其他tab页的对象上去，所以在不支持触摸屏的终端就上无法浏览所有tab页的内容，如图2所示。

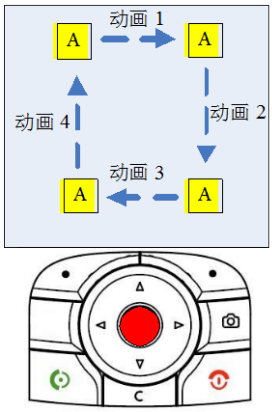

图1　动画循环执行示意图

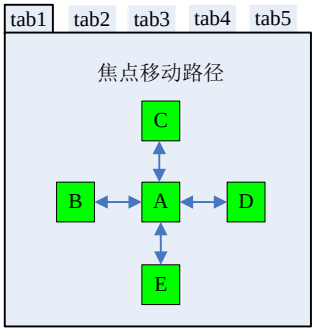

图2　多标签页焦点移动

原因是SVG Tiny 1.2只能通过鼠标点击重新设置焦点的位置，如果使用键盘则不能达到这种效果，这样对于不支持触摸屏的手机来说，应用开发有很大的局限性。

(3) 在一些富媒体应用中，当场景来回切换时，因为无法记住上一个场景的焦点位置，导致每次焦点都投放至同一个元素上，这样对一些例如体现多级菜单的应用场合十分不利，原因是场景切换时无法保存上一场景的焦点位置。

(4) 在一些特殊的富媒体应用中，例如动态界面，需要通过场景去调用第三方的应用程序，例如发送短信和拨打电话，如果支持脚本，则可以通过扩展API让脚本调用来完成此功能，但是如果富媒体引擎不支持脚本，则对这种需求无能为力。

2.2　改进方法

针对上述缺陷，通过分析脚本的用途，本文设计了一种用于在场景中实现逻辑控制的方法，变量操作以及终端本地能力的调用，配合SVG Tiny 1.2本身具有的动画能力及uDOM接口使用，可以达到很多应用设计的需求。

LASeR和DIMS的场景命令采用XML语言描述，其原因有二，一是XML语言表达的场景命令在执行速度上有着脚本无法比拟的优势，二是富媒体应用中需要通过更新数据动态创建元素的场合比较少，且这些工作可以放到服务端完成，所以LASeR和DIMS的一般做法是将服务端生成的更新场景传送至终端进行解析合成，最终渲染出来。参考LASeR和DIMS场景命令设计的思想，本文所述的方法也采用了XML语言的方式，这样做有很多好处。首先，从解析上不需要额外的模块支持，利用SVG引擎本身的解析器就可完成；其次，可以与SVG的DOM树紧密结合，因为在大多数SVG播放器的实现中都采用了DOM树的结构来存储节点的信息。

综上所述，本文所述方法具体的需求分为3个方面，即逻辑控制、动画操纵的执行以及终端能力的调用。

1) 逻辑控制方面需要满足：

(1) 创建一个变量并初始化，此变量可能用于判断条件中；

(2) 改变一个标量的值，此变量可能用于判断条件中；

(3) 将一个变量持久化或恢复，类似于编程语言的全局变量，可以跨场景使用；

(4) 停止某个命令的执行；

(5) 跳转到指定命令执行。

2) 动画操作需要满足：

(1) 触发场景中一个动画节点的执行；

(2) 终止场景中一个动画节点的执行；

(3) 重新设置场景中的焦点位置。

3) 终端能力的调用需要满足可扩展的方式调用终端的功能，如短信，浏览器等。

在具体的设计中，使用了XML标签对<cmldLine>来表示一条语句，每条语句可以完成上面3点中任意一子项的功能，每条语句带有一个判断条件，用于判断是否执行此语句。因为语句是作为XML节点存储于DOM树中的，根据树的存储结构可以很容易的实现语句的顺序执行。

下面一个语句的简单例子：

< cmdLine test=”b= =1” action =” urlcmd:// SetVar:a+1” />

< cmdLine ……../>

这条语句的含义是如果*b*等于1时，将变量*a*的值加1，否则跳过这条语句执行下一条语句。语句必须包含在容器标签中才有意义，而容器标签需要定义这些语句的执行条件，在SVG Tiny 1.2中，可以使用<script>标签和<handle>标签作为语句的容器，<handle>是SVG Tiny 1.2的特有标签，是DOM2接口addEventListener的XML表现形式，当响应的事件发生时，需要执行一次<handle>标签下的内容。因为<script>标签和<handle>都无法通过时间或者其他动画同步事件触发，所以本文为<cmdLine>定义了另外一种容器<cmd>标签，<cmd>标签的更像是带了begin属性的<script>标签，可以接受时间和多种事件的触发，包括SVG动画的同步事件。在结构上<cmd>标签可以包括一个或多个<cmdLine>标签，<cmd>标签的主要属性如表1所示。

表1 cmd标签属性

属性名称	属性描述	值	默认值
id	<cmd>标签的标识符	<ID>	N/A
begin	<cmd>标签的触发条件，根据SVG Tiny 1.2的定义，这个属性的取值可以是时间，UI事件或者其他动画同步事件[1]	“begin_value_list”	indefined

<cmd>的表现形式如下：

<cmd id=”c1” begin=”0s”>

<cmdLine……../>

…

<cmdLine……../>

</cmd>

例子的意思是当场景时间是0 s的时候，执行其下所包含的各个<cmdLine>语句。

<cmdLine>标签用于描述一条完整的语句，语句的内容是根据判断条件test是否满足去执行action属性定义的操作，标签主要包含两个属性，如表2所示。

表2 cmdLine标签属性

属性名称	属性描述	值	默认值
test	定义了是否执行后面action属性所包含内容的条件	condition	N/A
action	所包含内容的格式符合RFC2396的描述，但是其目的并不是去定位资源，而是借助url格式的描述能力去描述的命令或者动作。	<URI>	N/A，required

test属性用于判断<cmdLine>语句是否执行，判断条件可以是无条件执行或某变量的值是否等于、大于、小于、不等于另一个变量或某一数值。

test属性的值定义为condition，其描述为：

condition =var [">"|"<"|"=="|" != "] [var| decimal]

condition的具体表现形式例如：a>b，或者c！=5等。

action属性借助了URI去描述一些操作，这些操作根据之前的讨论分为3类：

1) 逻辑控制方面。

(1) urlcmd://SetVar?*a* =*a*+1 (变量 *a* 加 1)；urlcmd://SetVar?*c* =1 (创建一个变量 *c*，并且将其值置 1)；urlcmd://SetVar?*b*=*b*-2。(变量 *b* 减 2)；

(2) urlcmd://Save?*a* (保存变量 *a* 的值)，urlcmd://Restore?*a* (恢复变量 *a* 的值)；

(3) urlcmd://Return 退出当前的执行环境；

(4) urlcmd://Continue 返回到此<cmdLine>容器标签的第一个语句执行。

2) 动画操作，这类操作主要是调用 SVG 已有的 uDOM 接口函数，例如触发或结束动画的执行以及改变焦点的投放位置等，如果需要更多的操作可以方便的扩展。例如：

(1) urlcmd://Active? id=#animate，启动场景中 id 为 animate 的动画标签，Activate 可能会调用 uDOM 的 beginElement()接口；

(2) urlcmd://Deactive? id= #animate2，停止场景中 id 为 animate2 的动画标签，Deactivate 可能会调用 uDOM 的 endElement()接口；

(3) urlcmd://SetFocus? id= #rect1，设置场景中 id 为 rect1 的对象作为当前焦点，SetFocus 可能会调用 uDOM 的 setFocus()接口；

3) 终端能力调用，而且根据终端的不同可以扩展或裁剪所支持的功能。例如：

urlcmd://SMS?num=10086&content=ok, 调用终端短信功能向号码10086发送短信，内容为ok。

<cmdLine>语句的父节点可能是<handle>、<script>、<cmd>，但<cmdLine>标签不能包含任何子节点。需要注意的是虽然<cmdLine>语句可能作为DOM树的一部分存储，但是不能够通过uDOM接口对其进行访问。

当<cmdLine>语句被包含于<script>时，在整个场景文件解析的过程中就需要被执行，当被包含于<handle>时，根据<handle>的事件注册情况进行执行，当被包含于<cmd>结点时，根据begin属性定义的条件发生时执行。执行时顺序如下所示：

1) 读入第一个<cmdLine>语句。

2) 查询其包含的判断条件是否满足，若判断条

件不满足，执行步骤3)，若满足，则查询其操作类型。

(1) 若操作类型为“停止某个命令的执行”则执行步骤4)；

(2) 若操作类型为“返回某个命令的第一个子命令行执行”，则执行步骤1)；

(3) 若操作的类型为其他，则执行对应的操作后执行步骤3)。

3) 读入下一条子命令行，若存在，返回步骤2)；若不存在，返回步骤4)。

4) 退出此命令容器的执行周期。

2.3 实现

在实现上，本文通过扩展富媒体场景文件的解析器，类似xml解析器，使得在解析的过程中可以将<cmdLine>语句作为其容器元素的子结点存储在DOM树中，这样做的好处是当语句需要被多次执行的情况下不需要被反复解析。执行时按照上一章节的所述的执行流程进行执行。针对action的处理，可以利用正则表达式匹配将所需要的命令、属性以及值提取出来，进行处理并执行。针对2.1节所述的部分缺陷，采用本文的方法进行了实现。

(1) 实现了同一个事件根据条件的不同执行不同的动画。图3是采用类似方式实现的旋转菜单效果图。

a. 未点击之前　b. 动画执行过程中　c. 执行完成效果

图3　旋转菜单状态变迁示意图

```
<cmd id=”clickFire” begin=”accessKey(FIRE)”>
< cmdLine　　action =” urlcmd:// SetVar:a+1” />
< cmdLine test=”a==1” action =” urlcmd:// Active? id= #animate1” />
< cmdLine test=”a==2” action =” urlcmd:// Active? id= #animate2” />
< cmdLine test=”a==3” action =” urlcmd:// Active? id= #animate3” />
< cmdLine test=”a==4” action =” urlcmd:// Active? id= #animate4” />
< cmdLine test=”a==4” action =” urlcmd:// SetVar? a=0” />
</cmd>
```

(2) 实现了在不支持触摸屏的设备上，实现焦点设置的表示方法。图4是实现的tab页的效果图。

```
<cmd id=”choiceFocus” begin=” accessKey(FIRE)”>
< cmdLine test=”a==0” action =” urlcmd:// SetFocus? id= #tab1” />
< cmdLine test=”a==1” action =” urlcmd:// SetFocus? id= #tab2” />
< cmdLine action =” urlcmd:// SetVar? a = a+1” />
< cmdLine test=”a>1” action =” urlcmd:// SetVar? a=0” />
</cmd>
```

a. tab页1示意图

b. tab页2示意图

图4　富媒体tab页效果图

(3) 对于终端能力的调用以及变量的保存和恢复的例子，限于篇幅限制，在这里就不再举例了。

3 测　试

本文所述方法使得标签语言可以描述逻辑，操作变量，以及调用API，不需要复杂的脚本编程即可实现丰富的效果。但是与脚本的能力相比还有很大的不足，例如复杂的数据结构，面向对象的支持。但是其优点是可以大幅减小程序空间，执行速度快，在一些低端平台上优势明显。本文与当前流行的脚本引擎spiderMokey进行了对比测试，如表3所示。spiderMokey是执行javascript脚本的引擎，1.7版本，

用于firefox浏览器中，一些嵌入式浏览器也选用了它。

表3 本文方法与spiderMonkey功能对比

	本文所述方法	SpiderMonkey1.5	SpiderMonkey1.7
ROM Size/KB	12	260	370
RAMSize/KB	<50	>100	>100
分支判断	支持	支持	支持
循环	支持	支持	支持
变量操作	支持	支持	支持
接口调用	支持	支持	支持
复杂变量	不支持	支持	支持
面向对象	不支持	支持	支持

在性能方面，将集成了spiderMonkey的富媒体引擎与集成了本文所述方法的富媒体引擎在一些关键的条件下进行了对比，如表4所示。平台配置的cpu速度为416 MHz的PDA。其中页面切换速度包括了读取、解析和渲染的时间；调用终端API是从终端接受用户点击事件至执行具体的API之前经过的时间。

表4 本文方法与spiderMonkey性能对比

	本文所述方法	SpiderMonkey1.7
页面切换速度/ms	50	72
调用终端API速度/ms	5	12
调用uDOM接口速度/ms	5	12

在富媒体应用所需功能方面与目前流行的富媒体标准LASeR与DIMS在交互性方面进行了一些比较(不使用脚本的情况下)，如表5所示。

表5 本文方法与LASeR以及DIMS的对比

特点	本文所述方法	LASeR	DIMS
激活并执行动画	支持	支持	支持
停止并失效动画	支持	支持	支持
设置焦点位置	支持	不支持	不支持
切换不同焦点移动路径	支持	不支持	不支持
根据条件触发动画或动作	支持	不支持	不支持
调用终端能力	支持	不支持	不支持

4 结 论

本文提出了一种基于标签语言的逻辑控制方式，用于在一些场合中替代脚本的部分功能，这种方法使得SVG Tiny 1.2语言或者其他基于SVG Tiny 1.2的富媒体描述语言具有了简单的逻辑控制功能。扩展这种方法的目的并不是为了实现脚本的所有能力，而是在没有使用脚本的情况下以最小的代价实现更加复杂的表现逻辑，完成更多的功能，使得SVG Tiny1.2能够适应于更多的富媒体应用情景，提升用户体验，这种扩展方式不仅只能用于SVG Tiny 标准，也可以用于其他基于XML规范的标准中，例如XHTML，LASeR，DIMS，SMIL等。

参 考 文 献

[1] W3C. Scalable vector graphics (SVG) tiny 1.2 specification [S]. 2008.

[2] DUFOURD J C, AVARO O. An MPEG standard for rich media services[J]. Multimedia, IEEE JNL, 2005, 4(12): 60-68.

[3] 3GPP. Dynamic and interactive multimedia scenes (Release 7)[S]. 2007.

[4] SETLUR V, CAPIN T, CHITTURI S, et al. More: a mobile open rich media environment[C]//Multimedia and Expo, 2006 IEEE International Conference. Toronto: IEEE Press, 2006.

[5] MARILLY E, DELEGUE G, MARTIONT O. Rich media service creation for interactive mobile TV. Next Generation Mobile Applications, Services and Technologies[C]// NGMAST' 07 The 2007 International Conference. Cardiff: IEEE Press, 2007.

[6] YOUNUS A, SAMAD W A, STOCKHAMMER T. Dynamic interactive multimedia scenes in mobile broadcast environments[C]//ICC '07 IEEE International Conference. Glasgow: IEEE Press, 2007.

[7] LIM Y, JOUNG Y, CHEONG W. The simple aggregation format for lightweight applications scene representation (LASeR)[C]//ICCE '06 International Conference. Las Vegas: IEEE Press, 2006.

[8] SIGNES J. Binary format for scene (BIFS): combining MPEG-4 media to build rich multimedia services[C]// Proceedings of SPIE, the International Society for Optical Engineering. [S.l.]: SPIE, 1999.

[9] JOUNG Y, CHA J, CHEONG W S. An efficient type codec for point data in lightweight applications scene representation (LASeR)[J]. ETRI Journal, 2005, 6(27): 818-821.

[10] STECKEL P, SPIKA M. The hybrid java and XML-based MOBISERVE rich media middleware for DVB IP datacast[C]//Mobile and Wireless Communications Summit. Budapest: [s. n.], 2007.

[11] CONCOLATO C, LE FEUVRE J, MOISSINAC J. Design of an efficient scalable vector graphics player for constrained devices. Consumer Electronics, IEEE[J]. Transactions, 2008, 2(54): 895-903.

[12] KIM H. Supporting B2B business documents in XML web services[J]. Journal of Electronic Science and Technology of China, 2004, 2(3): 53-57.

编 辑 税 红