

第1章 VoiceXML的概述

1.1 VoiceXML语言

VoiceXML（语音扩展标记语言）是一种基于 XML 的因特网标记语言，用于开发语音用户界面。它是“语音 Web”使用的语言，它使得用户可以使用电话来访问因特网托管的内容。我们可以将其视为用于电话的 HTML。

使用语音界面，用户可以用他们的声音而不是通过计算机键盘和显示器与应用程序进行交互操作。在银行、学校或证券交易所，你可能已经通过语音邮件或者交互式语音应答（IVR）系统接触过语音界面。这些系统提示用户输入相应的信息，用户再根据要求输入相应的数据作为应答，然后系统就可以为用户执行有关的在线操作，例如，在不同账户之间进行资金转账。

在语音界面中，有一个 VoiceXML 对话框（大致相当于一个 HTML 页）用于描述应用程序对用户说出的提示信息，定义和接收来自用户的应答，并且描述程序控制的流程。另外，VoiceXML 还给电话规定了一些特殊的操作，例如，从一个 VoiceXML 应用程序转移到另一个电话呼叫的功能。

用户通过拨打应用程序的电话号码来访问 VoiceXML 应用程序。从用户的角度看，这个电话号码相当于一个 Web 网页的统一资源地址（URL）。用户可以通过任何类型的电话拨打，包括陆路电话、移动电话和卫星电话。

电话的类型将会改变语音通过电话网络的传输方式，但对于大多数应用程序开发人员来说，这并没有什么关系。我们将在第 2 章“运行 VoiceXML 的网络”中讲述这个问题。

为什么要创造一种新语言？

早在着手开发语音应用程序之前，VoiceXML 的设计者就决定创造一种新的标记语言，而不是扩充 HTML 的功能。这是因为采用语音的操作界面与 Web 浏览器一类的视觉显示方式有着本质的区别。

1.2 一个 VoiceXML 应用程序方案

要弄懂 VoiceXML 是怎样运行的，最简单的方法就是从头到尾将一个小型用户应用程序方案运行一遍。下面是一个简单的例子，我们将在全书中引用它。在我们的应用程序方案中，下面这个对话是在一个呼叫者与一个图书销售商的服务程序之间进行的：

服务器程序：欢迎光临 Acme Books 在线书店，你可以在任何时候说“Top Menu”（主菜单），以便返回本菜单。说“Info”，就可以听到我们书店的营业时间。说“Check out”（结账）就可以完成购买。说出书的名字，就可将书放入购物车中。

用户：Info。

服务程序：Acme Books在线书店全年365天、全天24小时营业。

用户：再见。

服务程序：感谢你访问 Acme Books在线书店。

下面是在更高层次上该在线书店的服务程序与用户进行谈话时必须采取的操作步骤：

- 1) 用户拨打在线书店服务程序的电话号码。
- 2) 负责托管该在线书店服务的语音服务提供商（VSP）的VoiceXML网关，接收到这个电话呼叫和该呼叫的有关信息。例如，被呼叫的号码（用户拨打的号码）和主叫号码（用户的电话号码）。
- 3) VoiceXML网关搜索一个数据库，并且将被呼叫的电话号码转换为一个统一资源地址（URL），这个URL就是Web服务器上的在线书店服务程序的 VoiceXML主页（books.vxml）的位置。
- 4) VoiceXML网关激活books.vxml主页，如果预先进行了设定，那么也可以激活相关的文件，例如，语法文件或录制的声音文件。相关的文件可以被高速缓存在 VoiceXML网关上。
- 5) VoiceXML网关对VoiceXML进行翻译，一步一步执行 books.vxml文件，并根据应用程序的规定与用户之间进行交互操作。
- 6) 根据需要，从Web服务器上下载其他的VoiceXML文件和相关文件。

1.3 语音浏览

根据World Wide Web联盟的“语音浏览器的活动”的报告，“语音浏览器允许每个人通过任何电话来享受基于Web的服务，从而使得人们可以在任何时间和任何地点，无论是在家中，在路上，还是在办公室，都可以访问Web。”

语音浏览器是驻留在网络上的一种应用程序，它位于语音、电话和因特网组件之上，负责将VoiceXML应用程序传递给用户。语音浏览器由负责解析 VoiceXML文档的VoiceXML翻译器以及连接语音和电话组件的界面组成。

由于语音浏览器的概念与HTML或无线通信技术相关，而浏览器又位于终端设备上，因此位于网络上的语音浏览器的概念似乎有点儿让人搞不清楚。在这里，语音浏览器与HTML和WAP（无线应用协议）很难进行比较。在下一节中，当我们把语音浏览器放在网络环境中来加以说明时，语音浏览器的功能就会非常明显地体现出来。我们还将介绍 VoiceXML网关的功能，它是托管语音浏览器的实际网络组件。而现在则让我们将注意力集中在浏览器的功能——浏览上。

语音浏览器的巨大潜力存在于无所不在的电话和独特的语音应用中。对于世界上的许多人来说，也许他们无法做到人人拥有一台PC机，但是在发达国家和发展中国家中，电话通常都被视为一种标准的家用电器设备。

在某些情况下，例如用户正在驾驶汽车时，如果他想要访问因特网，那么他只有使用免提电话和像语音那样的免提界面，才能非常安全和舒适地上网。例如，通过语音浏览器来查询交通路况信息，并不比你一边开车一边拨打任何其他电话更困难。相比之下，利用无线设备的图形浏览器来查询因特网的内容，则需要进行一定的身体和脑力上的训练，并且费用相当昂贵。

带有某些残疾的用户也可以很好地利用语音服务系统。由于没有考虑到用户的能力，有些

用HTML和WML开发出来的因特网站点和应用程序在使用时很不方便，因为这些服务更加需要正常的语言功能才能使用。这些内容一旦通过语音浏览器来显示，就可能吸引更广泛的听众。

许多内容提供商除了使用HTML和WML外，还用VoiceXML来传递他们的内容，目的是为了寻求最大数量的用户，就像他们匆匆忙忙地将无线服务添加到他们的传统因特网内容上去一样。

VoiceXML 的历史和演变发展

VoiceXML的起源可以追溯到1995年，当时AT&T公司研究部的计算机科学家开始着手Phone Web项目的开发。当朗讯公司脱离AT&T时，AT&T公司继续进行Phone Web项目的开发。这两家公司开发出不同版本的电话标记语言（PML）。最初的研究人员中有两个人进入了摩托罗拉公司，他们开发了VoxML。IBM公司也展开了类似的研究工作，从而导致SpeechML语言的问世。

这些公司成了VoiceXML论坛最初的4名成员，1999年该论坛开发了一种新的语言技术规范，即VoiceXML 0.9版。该论坛的发展十分迅速，2000年3月发布了VoiceXML 1.0版的技术规范。它被提交给World Wide Web联盟，该联盟的语音浏览器工作组将负责开发该技术规范的下一个版本VoiceXML 2.0。

当本书付印时，VoiceXML 2.0版的技术规范还是个草案，预计对VoiceXML 1.0版所作的修改大多与澄清有关的概念和纠正原版本的错误有关。本书将在第13章“VoiceXML的未来发展前景”中介绍技术成熟的VoiceXML 1.0版和即将发行的2.0版的变更。

1.4 VoiceXML在语音浏览器中是怎样工作的

一般来说，VoiceXML和语音浏览器通过为语音技术和内容传递提供标准界面，从而可以帮助编程人员开发因特网上运行的语音界面。尤其值得注意的是，VoiceXML提供了一个连接自动语音识别（ASR）、文本语音转换（TTS）、双音多频（DTMF）、呼叫处理以及其他技术的标准界面。

虽然VoiceXML在一定程度上降低了这些技术的复杂性，但是要编写出高质量的语音应用程序，你就必须了解它们的这些复杂性。在下面两个小节中，我们将对VoiceXML程序的输入和输出所使用的一些关键基础技术做一个概述。更加详细的内容将在第2章中介绍。

1.4.1 应用程序的输入

自动语音识别是一种使口语输入能够被程序所理解的技术。为此需要将口语中的单词分解成离散的声音小段，这种声音小段称为音素（phonemes）。然后将这些音素与应用程序开发人员或语音识别工具提供的词汇相比较。这不是可调整的或自然语言的识别方式。每给一个提示信息，VoiceXML应用程序定义一组有限的有效用户输入，其他用户的应答将不能被应用程序理解。语音识别可以通过专门的硬件或软件来实现。

双音多频（DTMF）系统通过电话网络中的硬件和软件对按键式电话的键盘输入进行翻译，实现音频播放。每个按键都能够产生一个可被识别的独一无二的音频。该系统使得用户可以使用电话键盘将数字信息输入VoiceXML应用程序，并且由程序负责接收和理解这些数字信息。

1.4.2 应用程序的输出

文本语音转换技术（TTS）能够将输出的文本转换成“语音”（实际上是一种数字音频格式）。这使得应用程序能够拥有灵活的输出方式——将开发人员没有预料到的事情告诉用户。TTS的功能与ASR基本上相反，它将文字转变为音素，这些音素具有相对应的数字声音。

另一种语音输出是用某种标准格式储存的录制的声音文件。这些声音文件格式包括 WAVE和脉冲编码调制（PCM）。这些声音文件在应用程序被启用以前已经被编写和录制好了。所有录制的语音输出都必须事先确定。因此，TTS是一种更灵活的输出方式。尽管TTS技术在过去的几年里有了显著的改进，但是商用TTS语音输出的声音往往仍然比较死板，并且有时很难听懂。

这些就是VoiceXML程序员与他们的用户进行语音交互操作时使用的基本技术。

VoiceXML还为电话网络本身提供了一个非常小的界面，称为“转移（transfer）”标签。它使得程序能够将用户从当前运行的VoiceXML应用程序转到另一个电话号码。为了理解语音浏览器和转移呼叫之类的功能，就需要对有关的网络做一个说明。

1.5 高级网络结构

VoiceXML是怎样使你能够编写语音应用程序的呢？仅仅介绍一下语言技术规范显然是不够的。尽管不了解网络是如何将内容传递给用户的许多深层情况，你仍然可以编写VoiceXML应用程序，但是，如果对网络有所了解，那么你就会更好地体会你的应用程序应该遵循什么样的规则，并且有助于你开发性能更佳的VoiceXML应用程序。

VoiceXML是在XML、HTTP和TCP/IP等现行数据网络标准以及公用电话交换网（PSTN）和综合业务数字网（ISDN）中的电话标准等基础上建立起来的。

语音Web是由PSTN、因特网上的VoiceXML应用程序以及因特网与PSTN之间的VoiceXML网关等组成的。VoiceXML网关负责托管可进行语音浏览的特定硬件和软件。这些资源中的某些部分，如ASR和TTS，可以位于不同的网络组件中，并且可以进行远程访问。

在一个语音浏览器会话过程中，用户的电话呼叫通过PSTN到达一个VoiceXML网关。根据用户拨打的电话号码，该网关从因特网上下载相应的VoiceXML应用程序，并且可能将该应用程序存入高速缓存。然后该网关逐步通过VoiceXML，按照应用程序的定义与用户进行交互操作。

图1-1显示了一个典型的语音网络。

以下是对网络示意图中涉及的组件的说明：

- 呼叫方的电话——呼叫方访问VoiceXML应用程序时使用的电话。
- VoiceXML网关——在PSTN和IP之间架起桥梁的网关，并且负责托管VoiceXML浏览器、语音硬件和软件，进行被叫号码与URL的映射。
- Web服务器——这是一个在本网络中托管VoiceXML应用程序的服务器。通过编辑HTTP服务器支持的MIME（多用途因特网电子邮件扩充协议）的类型，可以从任何Web服务器来传递VoiceXML。

下面是对该网络示意图中涉及的两个网络的说明：

- PSTN——公共电话交换网，也称为简易旧式电话服务网（POPS）。这是我们大多数人在家

中使用的一种电话服务网，它能够传送我们说的话和 DTMF 交互操作信息，例如 VoiceXML 网关播放的提示信息和对呼叫方说的话作出应答。

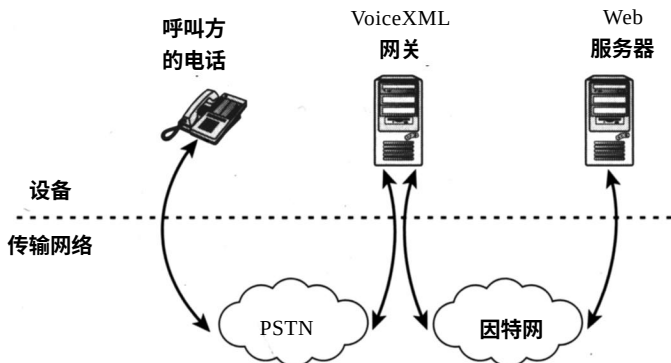


图1-1 VoiceXML网络组件之间的交互操作示意图

• Internet —— 我们向因特网服务提供商（ISP）支付费用以便对其进行访问的因特网。它将网关对VoiceXML的访问请求传递给Web服务器，然后将VoiceXML返回给网关。

请注意，该网络拥有一个相当传统的结构，不符合在因特网协议上传输语音（VoIP）的运行环境。这种结构类型将在第2章中进行讲解。

关于网络的一些重点概念可以归结为，语音Web是由PSTN、因特网上的VoiceXML应用程序和因特网与PSTN之间的VoiceXML网关构成的。VoiceXML网关负责托管实现语音浏览所需要的专用硬件和软件。

1.6 VoiceXML网关

作为一名VoiceXML开发人员，你的运行期平台是VoiceXML网关。明白了VoiceXML网关的运行方法将有助于提高你的VoiceXML网页的质量，调试也更加容易。随着你获得制作VoiceXML网页的经验，你也将会了解你的服务提供商所使用的VoiceXML网关的特性。

由硬件和软件构成的VoiceXML网关在PSTN与因特网之间架起了一座桥梁。这个网关包括一个VoiceXML浏览器和用于ASR、TTS和DTMF的资源，这些资源可能是硬件，也可以是软件。

下面列出了VoiceXML网关的主要功能：

- 电话 —— VoiceXML首先必须接收（也称为端结）来自公共电话交换网的电话呼叫并获取与呼叫相关的信息，比如被叫号码和主叫号码。根据呼叫人拨打的电话号码，网关可以为VoiceXML应用程序提供呼叫人的位置信息，例如呼叫人所在的城市和州。如果使用电话转接标记，将用户转接到另一个电话号码，那么VoiceXML网关负责启动一个连接到转接号码的新的外接呼叫，并且将它与用户的电话呼叫连接起来。
- URL数据库 —— 当网关接收到一个电话呼叫时，网关必须对被叫号码与要求的服务的URL进行核对。网关可以拥有与这个数据库相关的服务和计费系统。
- 检索VoiceXML —— 当知道VoiceXML服务程序的URL的时候，网关必须从服务程序的Web主机上检索VoiceXML页和相关文件，例如录制的声音文件和语法文件。

- 翻译VoiceXML — 当应用程序的VoiceXML代码和相关文件被存储在网关上之后，网关必须翻译这些代码，遍历对话框并且根据要求与 ASR、TTS、DTMF以及其他服务进行交互操作。这需要从Web服务器上检索其他的一些文件。
- 访问ASR和TTS — 这些服务可以作为软件或者硬件被托管在 VoiceXML网关上，也可以位于具备专用语音处理功能的远程服务器上。
- 高速缓存 — VoiceXML网关能够将预先录制好的声音文件、语法、以及 VoiceXML页存放在高速缓冲中。

IBM、朗讯、摩托罗拉和 VoiceGenie 等公司直接向服务提供商和大型门户网站销售商用 VoiceXML网关。其他公司生产的VoiceXML翻译软件可以与硬件和其他软件组件封装在一起，组成一个VoiceXML网关。一家名叫Speechworks的供应商则开放了它的VoiceXML翻译软件源代码。

1.7 托管VoiceXML应用程序

VoiceXML页可以像其他因特网文件那样由 Web服务器来托管。任何 Web服务器都可以使用，只要你能编辑MIME类型后缀映射表，在其中添加该文件类型：

```
Suffix: vxml Type: text/x-vxml
```

使用Apache、微软IIS和Sun公司的Java Web Server，你就能够很容易地修改 MIME类型后缀映射表，并且要托管 VoiceXML文档也不成问题。但是，仅仅将你的 VoiceXML应用程序放到一个Web服务器上是不够的，用户不一定能够访问它。在语音 Web上，只有当你的应用程序与一个电话号码关联起来之后时，用户才能访问该应用程序。有些地方 VoiceXML网关必须与你的应用程序相链接，这就必须有一个服务提供商。

VoiceXML服务提供商只具备因特网服务供应商的一半功能，即托管功能。虽然因特网服务供应商能够提供因特网接入和托管功能，但是语音 Web的接入功能则由无线和本地电话公司提供。

VoiceXML服务供应商（VSP）拥有托管语音服务的专业技术和设备，包括 VoiceXML网关、电话和因特网等基础设施，许多 VSP托管了它们自己的语音门户并提供咨询服务。这是 VSP开发的一种VoiceXML服务，用于访问VSP托管的实用应用程序或演示应用程序。

由于拨号进入VoiceXML网关的操作需要使用他人的资源，所以网关的拥有者可能要对用户扣留一定的费用，方法是向你收取访问费，或者向出现在门户上的网站收费。在目前这个施小恩谋大利的时代，很多公司将会免费托管你的 VoiceXML站点，有些公司甚至向访问站点的用户提供免费接入。

负责链接你的应用程序的VSP

VSP包括BeVocal (www.bevocal.com)、HeyAnita (www.heyanita.com)、Tellme(www.tellme.com)、VoiceGenie(www.voicegenie.com)和Voxeo(www.voxeo.com)，通过它们运行的开发人员程序，你可以进行注册并且免费通过它们的 VoiceXML网关来链接你的应用程序。

衡量一个VSP的优劣的方法是看他是否符合 VoiceXML的技术规范。W3C的语音浏览器协会

正在制定适用于 VoiceXML 浏览器的标准。另一个需要考虑的因素是它提供的附加的功能，例如它是否可以让开发人员使用位置信息。

1.8 选择 VoiceXML 应用程序

VoiceXML 应用程序呈现给用户的一系列对话框都是为了一个目的，即 WWW 用户目前在他们的图形浏览器上做的事情：获取信息、发送信息和购买产品。

尽管它们具有类似功能，但是出色的图形 Web 页不能直接转换成很好的 VoiceXML 页。语音用户界面向设计人员提出了一种特殊的挑战。当用户面对语音界面时，用户往往要求它具备比图形用户界面高得多的可用性。语音界面通常是非常出色的，否则它们就无法使用。第 4、5 和 6 章将要专门介绍语音用户界面的设计。

服务程序使用因特网只不过是有了一个驻留的位置。一个应用程序可能只是让用户浏览一个本地数据库。希望使用 VoiceXML 建立全新和独一无二的语音浏览器应用程序的开发人员应该综合利用语音和因特网访问技术。

无论你是编写 VoiceXML 应用程序来了解它是如何运行的，还是编写你公司的第一个语音应用程序页，你都应该将它们搞得对用户具有吸引力。选择正确的应用程序是很重要的。

下面是你在构思出色的语音服务程序时应该考虑的几个问题：

- 应该考虑用户要求或希望语音界面运行时所处的环境。什么样的数据是用户希望或需要的？
- 应该考虑什么类型的数据最适合用语音界面来显示。当用户访问那些数据时，他们会发现语音界面对他们有用吗？
- 你有现成的 HTML 和 WML 应用程序可以转换成 VoiceXML 应用程序吗？

1.9 用 VoiceXML 来开发应用程序

当你开始用一种新语言来开发应用程序时，观察一个常用应用程序实例将会对你有所帮助。每本关于编程语言的著作或教材上几乎都有一个“Hello World”应用程序的实例。

用 VoiceXML 编写的 Hello World 程序如下：

```
<?xml version="1.0"?>
<!DOCTYPE vxml SYSTEM 'voicexml.dtd'>
<vxml>
  <form>
    <block>
      Hello world.
    </block>
  </form>
</vxml>
```

VoiceXML 是一种比较小型的标记语言，目前它支持以下 47 个标记：

```
<assign>
<audio>
<block>
```

<break>
<catch>
<choice>
<clear>
<disconnect>
<div>
<dtmf>
<else>
<elseif>
<emp>
<enumerate>
<error>
<exit>
<field>
<filled>
<form>
<goto>
<grammar>
<help>
<if>
<initial>
<link>
<menu>
<meta>
<noinput>
<nomatch>
<object>
<option>
<param>
<prompt>
<property>
<pros>
<record>
<reprompt>
<return>
<sayas>
<script>


```
<subdialog>  
<submit>  
<throw>  
<transfer>  
<value>  
<var>  
<vxml>
```

由于 VoiceXML 是一种基于 XML 的语言，所以用 VoiceXML 来开发应用程序的方法与用 HTML 或 WML 开发应用程序的方法非常相似。使用大多数人阅读 HTML 指南和建立 Web 页的方法，他们可以很容易地开发出第一个 VoiceXML 对话框，它相当于一个 HTML 页。

用 HTML 编写 Web 页的方法已经被大多数人所掌握。但是，要开发一个设计优秀的网页或多分页网站就需要更多地了解基础技术和优秀界面设计的原则。用 VoiceXML 进行开发也是如此。

同样，从开发人员的角度来说，VoiceXML 之所以功能强大，因为它降低了基础技术的某些复杂性，并且可以通过一种他们熟悉的基于 XML 的高级语法来进行访问。利用 VoiceXML 和语音 Web，你能够迅速编写服务程序，而在其他的语言环境种这种工作是十分复杂的。

1.10 支持的语言和工具

如果你已经开发过因特网应用程序，你可能已经使用过诸如 ASP、Java 小服务程序或 CGI 脚本编程之类的工具。VoiceXML 支持这些服务器端的工具，并且支持 JavaScript/ECMA Script 直接在 VoiceXML 中编写功能更加完善的应用程序。大多数商业应用程序都需要许多这样的语言。

供开发人员使用的各种 VoiceXML 工具已经开始打入市场，这些工具的类型很广，从能够引入 DTD（比如 VoiceXML）的 XML 编辑器，到你可以在桌面计算机上用来测试 VoiceXML 的 VoiceXML 翻译器，无所不包。关于 VoiceXML 开发工具的详细介绍，请参见第 3 章的内容。

第2章 运行VoiceXML应用程序的网络

VoiceXML 应用程序需要一个“语音门户”，以便访问各种媒体资源，比如自动语音识别（ASR）和文本语音转换（TTS）功能。通过语音门户，VoiceXML应用程序就可以接入将服务传递给最终用户时使用的通信和因特网基础设施（见图 2-1）。

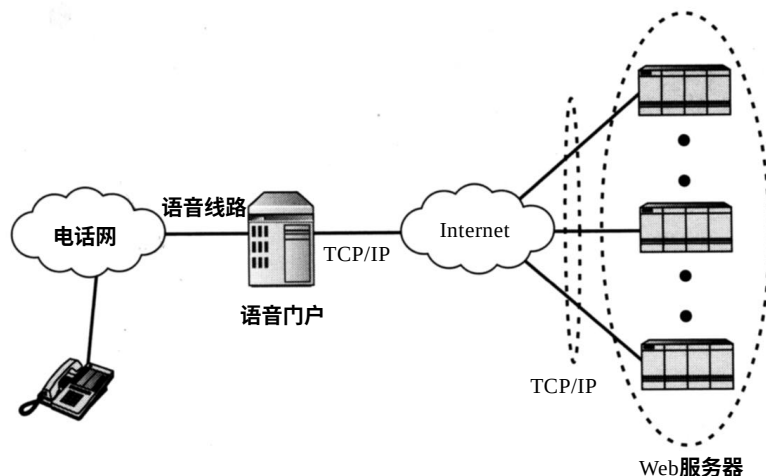


图2-1 运行VoiceXML应用程序的网络的主要组成部分，
它包括语音门户、电话网和因特网

如果要接入通信网，你既可以使用有线设备，也可以使用无线设备。有线接入是将最终用户的电话通过铜芯电话线（大部分新装用户可能是光缆）连接到通信交换机上。无线接入指的是使用各种无线技术，如全球移动通信系统（GSM）、码分多址（CDMA）和时分多址（TDMA）技术，建立与通信交换机的连接。请注意，在VoiceXML应用程序环境中进行的通话可以有电话与无线电话的混合。许多新型网络也支持IP电话的通用技术，在因特网上进行电话通信。

尽管使用这些网络拓扑后，新的电话服务提供商可以用较低的成本打入整个服务领域，但它们给VoiceXML网络和应用程序开发也带来了一系列的难题。我们将在本章后面的VoiceXML的应用程序结构的内容中介绍关于有线和无线网络拓扑的问题。不过在我们讲述这些内容之前，首先让我们介绍一下VoiceXML网关或语音门户的概念。

2.1 VoiceXML网关

VoiceXML网关或语音门户是运行VoiceXML应用程序的网络的心脏。图 2-2说明了它的组成结构。在向用户提供TTS、语音识别、媒体资源以及对因特网的访问功能的同时，VoiceXML应用程序通过语音门户来访问电话设备的用户。有关VoiceXML门户的电话功能的详细信息将在讨论电话资源时介绍。现在我们将要介绍门户的语音处理特性，它有助于正确判断与门户进行交

互操作时用户的感受。

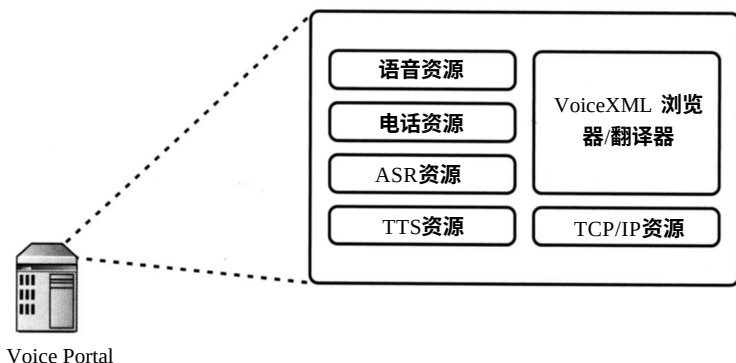


图2-2 语音门户的组成结构

2.1.1 TTS资源

TTS技术是将文本数据流分解成可发音的音素，再合成为人类语言的过程。拼音字母表由录音室录制的声音产生；TTS引擎使用特殊算法将这些声音合成为人类可辨认的单词和短语。有些供应商采用语音卡上的固件来实现TTS引擎的功能。其他供应商则选择使用基于主机的解决方案，以便实现高吞吐量，并且使不同的TTS引擎可以更加密切地纳入该平台。

TTS的播放请求在一个TTS通道上被接收，文本以数据流的形式顺序存入该通道上的一个缓冲区。如果出现很长的提示，那么由于有了这个缓冲区，因此可以连续播放文本数据流。由于文本数据流被转换成它的语音表达形式，因此可以通过电话资源将它传递给最终用户。用户通过双音多频（DTMF）输入或者语音输入来中断提示信息的播放，这种功能称为打断功能（Bargein），起着非常重要的作用。有效的打断方案的目的是要最大限度地缩短最终用户请求停止播放提示到系统实际停止播放提示之间的时间。打断时间大约为几百毫秒是常见的。

由于大多数语音门户结构均通过电话资源来支持DTMF，因此通过这个界面来传递TTS的输出很容易支持有效的打断机制。但是，对语音打断机制的支持在大多数情况下不如DTMF打断机制那样有效。语音打断机制要求有一个活动ASR资源和TTS资源连接到相同的端口。连接到相同的端口后，当提示播放时，用户发出的语音命令就可以“立即”被识别。当用户的语音命令被认定为有效命令时，语音处理器就被告知停止提示。

就像打断功能对于语音门户服务来说是衡量服务质量的重要因素一样，用户听到的TTS提示的质量，是决定他们使用语音门户应用程序的感觉的一个重大因素。提示信息可长可短，但如果提示的声音不好听，那么服务供应商就可能失去客户。

设计时应该考虑的一些问题：

当你进行TTS操作时，请牢记下列设计注意事项：

- 因供应商的不同，TTS的播放质量也差别很大。因此你应该向语音门户主机索取你的应用程序要采用的所有语言的TTS输出示例。正如以前介绍的一样，最终用户听到的声音质量将会影响他们对VoiceXML应用程序的总体感觉。

- 通过供应商的专用语音和 / 或文本控件, 可以对 TTS 的质量进行准确的调整。但是, 由于 VoiceXML 提供的是抽象技术, 在使 TTS 的输出能让大多数最终用户接受的各个控件之间并不存在一一对应的关系。因此, 语音门户供应商应该能够对他们的 VoiceXML 翻译器工具添加专门的扩展功能。
- 某些应用程序, 如 e-mail 阅读程序, 需要一个文本预处理器, 以便在将文本发送给 TTS 资源之前对文本进行格式化。当你设计此类应用程序时, 你应该了解一下是否可以将该功能添加给有关的语音门户。
- 对于传递给 TTS 资源的字符串的长度, 不应该规定任何的“限制”。你应该进行相应的测试, 以便确定 VoiceXML 引擎供应商是否能处理诸如在新闻阅读器应用程序中存在的超长提示。
- 当你选择一个平台时, 首先应该考虑它是否支持 DTMF 或者语音打断机制。用户希望打断机制的响应时间应该是几百毫秒或更少。

2.1.2 ASR 资源

ASR (自动语音识别) 是指计算机系统接收人的语音输入并返回对应于该语音输入的单词、短语或数字的过程。使用 ASR, 计算机能够识别来自一组相关的随机说话人说出的单词, 所谓“相关的”一词是指某种特定语言的语音模式常常是建立在具有代表性的讲这种语言的本地人的基础上。

注意: 通过训练, 语音识别器可以被调整为能够识别某个特定人物的声音。这种训练特性最适合支持语音拨号或者根据名字进行目录查找之类的功能。

与 TTS 资源一样, ASR 资源也可以用固件或基于主机的结构来实现。从最简单的概念上讲, 语音识别需要建立一个对象, 以便处理语音资源上的应用程序与语音识别引擎之间的界面。基本的语音识别请求的处理过程可以用程序清单 2-1 中的 VoiceXML 代码段以及它后面的说明文字来加以描述。假设呼叫者已经与语音门户相连接, 从而导致一个解释线程的实例得以建立, 然后便开始进行交互操作。

程序清单 2-1 用于语音识别环境的代码段

```
<?xml version="1.0"?>
<vxml version="1.0">
  <!--! This is a simple example of multiple active grammars that will demonstrate
  the Client-Server ASR Architecture. -->
  <form>
    <prompt bargein="false"> Hello, and welcome to the Amazon.com online
    book and music store.
    What would you like to buy, books, music?</prompt>
    <help>Say books or music.</help>
    <field name="item-choice">
      <grammar type="application/x-jsgf">
        books | music
      </grammar>
    </field>
  </form>
```

```
<prompt> I believe I heard you say <value expr="item-choice"/> . </prompt>
</filled>
</field>
</form>
<!--! At this point in the code, we will have activated the "help" grammar
and the inline grammar corresponding to "'books' or 'music'"-->
.
.
.
</vxml>
```

下列操作步骤提供了一个简化的语音识别处理过程的运行环境。这个运行环境可以用于程序清单2-1：

1) TTS资源以文本转换为语音的形式，显示提示“Hello, and welcome to the Amazon.com online book and music store. What would you like to buy, books, or music?”（你好，欢迎光临亚马逊在线图书与音乐商店。你想买什么，图书还是音乐？）。

2) Jspeech Grammar Format (JSGF) 直插式语法“books | music”使得语音识别客户程序能够建立一个代表该语法的语法对象。该语法包含了该语音识别程序实例能够为之返回有效结果的词汇表。如果用户说的单词或短语不是“图书”或“音乐”，那么语音识别器返回的信息是“没找到”或识别失败。

3) 对语法对象的引用被传递给语音识别服务器，以便进行编译和优化。

4) 呼叫者说话以后（或者接收到超过某个阈值的噪声），他们的声音修整后传递给语音识别服务器，对声音的修整可以采用噪声过滤或回声删除的方式。

5) 语音识别服务器接收音频输入，同时，由于它拥有一个对当前语音识别请求的引用，因此它将设法寻找匹配的单词“books”或“music”。

6) 假设所说的单词是“books”，并且语音识别服务器成功地找到了匹配的单词，那么，单词“books”将返回给翻译器线程实例。

随着人们对高吞吐量、多线程处理功能需求的增长，对电话语音识别引擎的需求量也在迅速上升。语音产品供应商，如 Nuance 公司，已经开发出一种客户机—服务器 ASR 结构，从而可以在一个或者多个语音识别服务器上以多线程形式分配语音识别请求。

2.1.3 VoiceXML 浏览器和翻译器

VoiceXML 浏览器和翻译器可以视为语音门户的“心脏”。从这点出发，浏览器和翻译器相结合就可以称为 VoiceXML 引擎。VoiceXML 浏览器提供了先进的类似因特网浏览器的功能，如书签功能，cookie 和高速缓存功能等，而翻译器则提供了对 VoiceXML 页的元素和属性的翻译功能。因此当今许多 VoiceXML 应用程序都紧跟因特网浏览器的发展方向。

当接收到一个启动事件，例如从外面拨入的电话呼叫时，浏览器通常首先使用 HTTP 协议提取该应用程序的第一个 VoiceXML 文档。如果浏览器的高速缓存功能已被激活，那么高速缓存的网页版本很可能就是该第一个 VoiceXML 文档页。第一个 VoiceXML 应用程序文档被浏览器安装在内存中，以便由翻译器进行翻译。元素和属性经过翻译而产生的操作，最终将驱动 ASR、TTS

和电话资源的运行。这些操作可以通过专用界面来启动，也可以通过更“标准”的界面，例如用于ASR和TTS的Java Speech API (JSAPI) 和用于电话的Java Telephone API (JTAPI) 来启动。

设计时应该考虑的一些问题

当你挑选一种VoiceXML引擎时，请记住下列设计方面的注意事项：

- 你要考虑的主要问题是确保 VoiceXML 引擎的实现程序能够始终符合当前的 VoiceXML 技术要求。供应商应该有一个他们的工具支持的元素和属性的清单。这时，你应该核实供应商作出的关于它们的引擎符合 VoiceXML 技术要求的任何声明。值得指出的是，截止到撰写这本书的时候，VoiceXML 论坛的一致性小组正在积极开发一个一致性软件，用于“检验”VoiceXML 引擎是否符合技术规范。
- 某些引擎供应商为他们的 VoiceXML 引擎添加了许多专用扩展特性。因此，你必须根据它们给你的应用程序增加的价值和 VoiceXML 技术规范的发展方向来衡量这些扩展功能。

2.1.4 声音资源软件

音频资源负责播放通常从因特网下载的声音文件。它支持的文件格式完全是根据不同的平台而定的，但是 .wav 格式的各种变形是宣传最广泛的支持格式。支持流式音频正在成为人们最希望获得的功能，特别是在通过电话传输新闻和音乐的市场领域中，更是如此。

设计时应该考虑的一些问题：

当你使用声音资源时，请牢记下列设计方面的注意事项：

- 根据最新统计，目前存在许多 .wav 文件格式类型的变形。你应该确定本机（也就是板级上）支持哪些文件格式，并且确定通过接近实时转换的子系统能够支持哪些文件格式。后面这个支持特性将对应用程序的吞吐量有着重大的影响。这些相同的问题也适用于其他音频格式。
- 截止到撰写本书时，支持流式音频特性还不是 VoiceXML 技术规范的组成部分。但是你需要考虑的问题是，你应该使用类似基于 PC 的媒体播放器那样的方式来控制媒体流的播放。该播放器应该考虑配备暂停、倒带、停止等功能。

2.1.5 TCP/IP 资源

TCP/IP 资源负责提供语音门户与它连接的 LAN 和 WAN 之间的骨干连接。它还负责语音门户与网络上的其他设备之间的数据包组装 / 分解的管理。由于你无法对 TCP/IP 资源实施控制，因此就不存在设计时需要考虑的注意事项。

2.1.6 电话资源

电话资源是语音门户的最重要的资源之一。电话资源负责提供门户与电话网络之间的界面。通常情况下，语音门户平台的开发人员既可以使用专用的 API，也可以使用更加“标准”的 API，例如微软的 TAPI（电话应用编程界面）或太阳公司的 JTAPI，将他们的 VoiceXML 浏览器 / 翻译器连接到电话资源上。从语音门户的角度来看，电话网络是从网络交换机开始的。包含网络交换机的电话网络将在下面进行详细的介绍。

在网络交换机与语音门户之间交换了一系列协议消息后，电话资源便在它们两者之间建立起语音连接。正是这个电话资源，为某些结构提供了回声删除、DTMF音频检测和语音提示播放/录制等功能。回声删除功能负责将来自语音信道的许多背景声音的回声去掉，使得用户听到的和语音处理器接收到的语音信号更加清晰。DTMF音调是大多数电话用户都很熟悉的“按钮音”。

这里值得注意的是，某些电话资源的结构是通过电话卡进行音调检测，而其他结构则通过语音处理资源中的支持特性来进行检测。语音提示是预先录制好的可以向最终用户播放的提示。另外，语音提示也可以进行录制。由于电话信道的带宽和语音传递信道的性能有限，向最终用户播放的语音提示的质量有着很大的不同。语音提示的播放和录制也可以用电话卡或者语音处理器卡来实现。

2.2 电话网络的拓扑

今天，电话是世界上使用最普遍的人造通信设备。传统上，人们一直使用有线（即 POPS、线路交换等等）网络来拨打他们的大多数电话。但是，越来越多的人正在同时使用有线电话网和无线电话网来拨打他们的日常电话。随着有线电话服务的商品化和无线电话服务竞争的加剧，导致无线电话服务的价格迅速下降。更多的人选择无线电话作为他们惟一的电话设备。本章的剩余部分将专门用来概述今天的许多 VoiceXML 服务使用的两种主要电话网的拓扑。

2.2.1 有线电话网的拓扑

为了部署 VoiceXML 服务的需要，有线网络可被分为触发式和非触发式两种。触发式网络服务是指呼叫者拨入本地交换机（服务交换点 SSP？）的号码将决定发送到本地交换机的服务控制点（SCP）的呼叫事件的类型。SCP 可以作为一个“智能”电话处理数据库服务器，通过一系列消息向交换机提供呼叫处理和路由指令。信号转接点（STP）的作用是在各个网络组件（即我们这里所说的 SSP 和 SCP）之间转发 7 号信令系统（SS7）信令。

注意：交换机也支持根据呼叫者拨打的号码而产生的事件。鉴于今天的电话用户群体具有很大的移动性，因此通常建议不要对语音门户结构使用基于呼叫号码的电话事件触发器。

这里需要考虑的一个重要问题是，当交换机与 SCP 进行通信时，这个呼叫“实例”将保留在交换机上，从而使交换机能够对电话进行完全的控制。但是，当交换机从 SCP 那里收到它的路由指令，并且将电话转发给语音门户以后，交换机便删除该呼叫实例，并且由语音门户保持对电话的完全控制。图 2-3 显示了一种典型的触发式有线电话网拓扑的结构图，而它所附带的电话呼叫实例则有助于我们更好地理解一些关键的概念。

下列步骤讲述了一个高级触发式拨号电话呼叫的例子。这个例子可以用于说明图 2-3 中的情况。

- 1) 呼叫者拨打电话号码 1-800-555-3932。
- 2) 由于 SSP 将号码 1-800-555-3932 定为被叫号码触发器，因此 SSP 根据该被叫号码向 SCP 发出一个查询。
- 3) SCP 对查询的内容进行分析，并确定这个电话应该转发给号码 919-555-8080 指定的语音门

户。随后 SCP 将路由指令发送给包含语音门户电话号码的 SSP。

4) 接收到来自 SCP 的路由指令后, 通过建立一个到达线路 919-555-8080 上的门户的主速率界面 (PRI) 链路, SSP 将当前电话转发给语音门户。该链路将呼叫方直接与语音门户相连接。

5) 这时, 交换机不再对电话进行任何控制, 语音门户可以开始处理 VoiceXML 页。

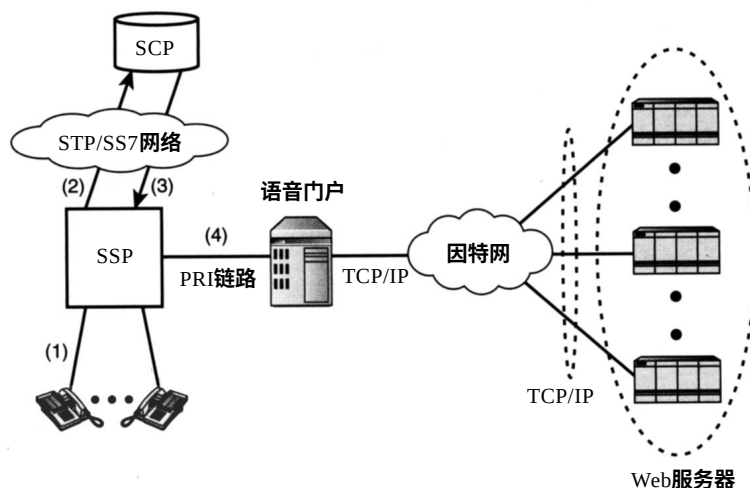


图2-3 触发式有线电话网的拓扑, 这种网络也是 8xx、
Caller ID 等高级智能网服务的基础

Telecordia 公司开发的高级智能网络 (AIN) 协议标准定义了一组消息, 从而使得它也能够支持上面所说的语音门户网络拓扑。请注意, 被叫号码 800-555-3932 被有效地转换成语音门户号码 919-555-8080。SCP 使得网络提供商能够在电话到达语音门户之前, 对外面打进来的电话进行其他的高级呼叫处理, 比如对电话进行筛选, 当拨打 8 字头的号码时, 呼叫方通常不必支付长途电话费。

非触发式有线电话采用类似的呼叫流程, 不过它不再出现 SCP 交互操作的过程。下面介绍的情况提供了一个高级非触发式拨号呼叫的例子。这个例子也可以用图 2-3 来加以说明, 当然它不包括与 SCP 的交互操作:

1) 呼叫方拨打电话号码 919-555-8080。

2) SSP 接收到打进来的电话, 然后通过分配给该号码的 PRI 链路将电话转发给语音门户。该链路直接将呼叫方连接到语音门户。

3) 这时, 交换机不再对电话进行控制而, 语音门户则开始处理 VoiceXML 页。

尽管触发式有线网络拓扑的功能要小一些, 但是因为它不需要 SCP, 所以部署语音门户的初始成本比较低。这种拓扑可能会被小型本地服务提供商或语音服务提供商 (VSP) 采用, 尽管它们无法访问 SCP, 但是它们的语音门户与本地交换机之间可以直接进行连接。

设计时应该考虑的一些问题:

当你使用有线网络拓扑时, 请牢记下列设计方面的注意事项:

- 虽然 VoiceXML 开发商无法控制已经部署的 VoiceXML 服务与现有电话用户服务之间的交互操作, 但是这仍是他们应该关心的一个问题。例如, 假设你的服务是根据电话用户的被

叫号码来提供的。另外，假设你电话用户已经拥有一种根据某个人拨打他的电话号码（比如CallerID）来提供的服务，它将呼叫方的姓名和号码传递给电话用户的 CallerID框。这两种服务都是建立在拨打相同号码的基础上的，并且从技术上是同时被激活的。因此，你必须确保语音门户供应商使用一个中介应用程序，以防止某一种服务打断另一种服务的运行。

- transfer（转接）元素也能够影响语音门户的功能，因此也影响到它部署的应用程序的功能。VoiceXML 1.0技术规范定义了两种类型的转接：盲发转接和桥式转接。盲发转接会将打进来的电话与打出去的电话连接在一起，然后转交给 SSP，而不考虑转交的方法。这一系列事件会导致语音门户失去对该电话实例的全部控制。而使用桥式转接时，打进来和打出去的电话也连接在一起，但是仍然保持了对当前 VoiceXML会话的控制。盲式转接产生的主要功能是，当提供服务的 SSP和/或语音门户不支持 2B信道转接功能时，它能够将电话顺利地转交给 SSP。如果不支持这个功能或类似的功能，或者如果在某些条件下，转接请求失败了，那么这个电话呼叫将被允许进行处理。在这种情况下，呼叫期内打进来的电话和打出去的电话各占一条独立的 PRI线路。因此进入语音门户的两条线路都将被占用。对于上面描述的网络结构来说，这就是盲式转接的预期结果。Transfer元素可能产生的另一个问题是如何进行电话计费，这个问题下面将要介绍。
- 虽然电话网络对于 VoiceXML设计者来说是透明的，但是 transfer元素的使用可能会在计费方面带来一定的问题。电话一般是向打电话的一方收费的，当然如果电话是免费电话，则另当别论。但是，对于本章中介绍的电话网络结构来说，将为打出去的电话生成一个新的呼叫实例，并且产生一个新的费用记录。因此，在使用元素时必须格外小心，因为打电话的呼叫方将电话转接到的这个电话号码，将根据其位置按长途电话号码来计费。你应当弄清楚你的语音门户提供商支持什么样的收费方式。要搞清楚这些电话是应该向打电话的一方收费，还是向被呼叫的一方收费，或者向服务提供商收费。

2.2.2 无线电话网的拓扑

在提供通信服务方面，无线电话的作用正在与日俱增。今天使用的无线电话网主要有 3种，即GSM、IS-41和PCS。不过，为了简明扼要地说明问题，本章只介绍 IS-41标准电话网。若要了解详细情况，请参阅Travis Russell撰写的《Signaling System # 7》第3版，该书由 McGraw-Hill出版社于2000年出版。

IS-41技术标准是由美国电子工业协会（EIA）开发的，该标准在技术概念上与 AIN标准有着若干相同之处，它们都使用 SS7信令，在某些网络组件之间传递消息。在 IS-41网络中，本地位置注册表（HLR）起着—个数据库的作用，用于存放供应商的本地服务区域的所有电话用户的信息。HLR数据库既可以与移动交换中心（MSC）放在同一个位置，也可以是一个远程数据库，有时也称为独立本地位置注册表（SHIR）。MSC是无线电话网中相当于有线电话网中 SSP的机制，用于建立无线网络内部的连接以及与有线网络的连接。

当使用漫游功能时，访问者位置注册表（VLR）用于存放不在本地位置的无线用户的访问信息。VLR数据库也用于存放访问者的移动电话号码，这样访问者就可以在网络覆盖的其他地