



Internet 新技术丛书

VoiceXML

语音应用程序开发

*Voice Application
Development with
VoiceXML*

(美) Rick Beasley 等著

王建华 王卫峰 等译

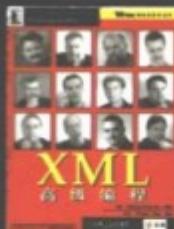


机械工业出版社
China Machine Press

SAMS

Voice Application Development with VoiceXML

VoiceXML 语音应用程序开发



《XML 高级编程》
ISBN 7-111-07315-0
944页 / 95.00元



《微软 XML 解决方案》
ISBN 7-111-08840-9
294页 / 46.00元



《XML 学习指南》
ISBN 7-111-08622-8
254页 / 40.00元

ISBN 7-111-09777-7



9 787111 097778

VoiceXML是怎样运行的?

以下是一个简单的例子——以下情景发生在一个手机用户与一个图书销售商的服务程序之间:

- 服务程序: 欢迎光临, 您在任何时候说“主菜单”, 可以返回本菜单; 说“简介”, 就可以听到我们的营业时间; 说“结账”就可以完成购买; 说出书名, 就可以把书放入购物车。
- 手机用户: 简介。
- 回应道: 本在线书店全年365天、全天24小时营业。
- 用户说: 再见。
- 服务程序: 感谢访问本在线书店, 欢迎下次光临, 再见。

这一切只需通过手机就可实现, 而VoiceXML正是关键所在。

已出版参考书:

- 《XML技术内幕》
- 《XML速成教程》
- 《XML技术大全》
- 《 XHTML技术内幕》
- 《 XHTML初级编程》
- 《XSLT技术内幕》

适用水平: 中、高级



华章图书

www.china-pub.com

北京市西城区百万庄南街1号 100037

购书热线: (010)68995259, 8006100280 (北京地区)

总编信箱: chiefeditor@hzbook.com

ISBN 7-111-09777-7/TP · 2296

定价: 35.00 元

Internet新技术丛书

VoiceXML语音应用 程序开发

(美) Rick Beasley 等著
王建华 王卫峰 等译



VoiceXML是一种基于XML的因特网标记语言，可以用来开发语音用户界面，使人们可以通过移动电话来访问因特网。本书通过VoiceXML的基本概念、VoiceXML的用户界面、VoiceXML应用程序的开发过程、VoiceXML的基本元素和属性以及VoiceXML的高级开发技术等专题，对VoiceXML应用程序的开发进行了详细的介绍。

本书内容翔实、深入浅出、极具参考价值，既适合VoiceXML应用程序的开发人员使用，也适合任何希望了解新型因特网访问技术的人员阅读。

Rick Beasley, et al: Voice Application Development with VoiceXML.

Authorized translation from the English language edition published by Sams, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2001 by Sams. All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2001-4286

图书在版编目（CIP）数据

VoiceXML语音应用程序开发 / (美) 贝斯利(Beasley, R.)等著；王建华等译。—北京：机械工业出版社，2002.2

(Internet新技术丛书)

书名原文：Voice Application Development with VoiceXML

ISBN 7-111-09777-7

I. V… II. ① 贝… ② 王… III. 可扩充语言，XML—程序设计 IV. TP312

中国版本图书馆CIP数据核字（2002）第001151号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：梁开莉 张鸿斌

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002年2月第1版第1次印刷

787mm×1092mm 1/16 · 18印张

印数：0 001-4 000册

定价：35.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

近年来，移动通信技术得到了空前迅速的发展，移动电话的用户出现了爆炸式的增长。有人估计，到2005年，全球将有10亿部手机投入使用。由此人们对使用移动电话在任何时间和任何地点访问因特网产生了浓厚的兴趣，并且这已经成为许多人的普遍愿望。可以预言，届时，利用语音技术通过移动电话来访问因特网将会成为一种非常流行的手段。

VoiceXML（语音扩展标记语言）是一种基于XML（扩展标记语言）的因特网标记语言，它可以用来开发语音用户界面。通过语音用户界面，用户就可以通过电话来访问因特网上的内容，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。为此我们可以将VoiceXML视为用于电话的HTML语言。

本书是专门为了满足从事VoiceXML应用程序开发的广大编程人员的需要而编写的。全书分几个专题对VoiceXML应用程序的开发进行了全面的介绍。第1~3章介绍VoiceXML的基本概念，讲述你开始学习VoiceXML时首先需要掌握的基本知识。第4~6章介绍VoiceXML用户界面(VUI)，说明与VoiceXML的用户界面有关的各种问题。第7章介绍VoiceXML应用程序的开发过程以及需要考虑的有关问题。第8、9章介绍VoiceXML应用程序的开发基础，详细讲述了VoiceXML的基本元素和属性。第10~12章介绍VoiceXML的高级开发技术，分别讲述支持访问特定平台的函数、ECMAScript的基本概念、ECMAScript与VoiceXML的集成、进行语音录制和电话呼叫转接时使用的各种元素，以及用VoiceXML进行服务器端编程的方法。第13章对涉及VoiceXML的未来发展的某些问题进行了展望。

本书配有3个附录。附录A提供了一个VoiceXML 1.0版的元素列表，对每个元素的功能进行了简要的介绍。附录B给出了本书开发的完整的在线书店应用程序。附录C简要介绍了VoiceXML 2.0版对1.0版可能进行的修改。

本书既适合从事VoiceXML应用程序开发的软件技术人员使用，也适合任何需要或希望了解新型因特网访问技术的人员阅读参考。

本书由王建华、王卫峰、杨惠琼、叶兆民、张晓佳等翻译，席赛珠校对，王卫峰、芳妮录入。由于译者水平所限，不妥之处敬请读者指正。

2001年11月

AMIS07/07-06

前　　言

“到2005年时，随着10亿部手机投入使用，语音肯定会成为你离开办公桌后访问因特网时所使用的最流行的（尽管不是最普及的）手段。VoiceXML是一种将电话与电信经营商、服务提供商、开发商和消费者联系在一起的工具。”（引自Minde公司的Web站点，网址是<http://www.minde.com/solutions/vxml.html>。）

目前，用移动电话访问因特网看来已经成为每个人的普遍愿望。在任何时间、任何地点访问因特网的梦想，使人们对此产生了浓厚的兴趣，并且希望尝试使用如VoiceXML和WML（无线标记语言）等瘦客户移动支持技术来访问因特网。尽管在美国很少采用像WML那样的小型屏幕浏览器访问技术，但是，由于人们始终希望采用移动式因特网访问技术，因此人们对VoiceXML和语音门户技术产生了更加浓厚的兴趣。

人们对以非PC为基础的移动电话访问因特网的技术的兴趣始终未减。如果你正在开发因特网应用程序或服务，那么你将很快有机会使用瘦客户机访问方式（如果你还没有使用的话）。

如果你已经在开发因特网服务程序，可能会想到VoiceXML简直是“妙极了！它完全是另一种*ML（标记语言）”。虽然VoiceXML是一种比较易学的标记语言，但你很快会发现语音应用程序具有许多开发人员前所未遇的复杂性。本书将有助于你学习如何解决语音应用程序开发中你不熟悉的一些问题，包括它们的支持结构和使用的平台、开发过程的不同、VoiceXML的元素和属性、语法和VoiceXML的用户界面。

本书的读者对象

本书编写的目的为了满足广大从事或涉及因特网相关技术开发的人们进行学习的需要。这些人包括：

- 为VoiceXML应用程序进行结构设计或制定技术要求的工程师。
- 从事VoiceXML应用程序开发的软件开发人员。
- 为VoiceXML应用程序制定和执行测试方案的软件测试人员。
- 负责设计VoiceXML应用程序的提示、消息或帮助信息的人类行为工程学家或其他专家。
- 需要了解VoiceXML应用程序的功能和结构的销售人员和市场专业人员。
- 当因特网服务提供商开始托管VoiceXML应用程序时，那些希望首先利用这些应用程序的业余爱好者。
- 任何需要或希望了解重要的新型因特网访问技术的人。

要想从本书中获得最大收益需要具备哪些条件

若要从本书中获取最大的收益，你需要具备的条件如下：

- VoiceXML的说明书，这是你了解VoiceXML元素和属性信息的最佳参考资料，你可以从VoiceXML论坛的Web站点上获得该说明书，它的网址是<http://www.voicexml.org/>。
- 能够运行你希望使用的任何专用软件开发工具包（SDK）的计算机平台（参见软件开发工具包供应商的说明书），它能够提供基本的因特网访问工具，并且可以使用文本编辑器或者XML编辑器来编辑VoiceXML文件。
- 能够访问模拟器、测试环境或语音浏览器平台。有关VoiceXML工具和对开发人员的支持特性的详细信息请参见第3章。
- 具备电话接入功能，这样你就可以通过拨号呼叫对环境进行测试或者建立环境的原型。

本书中的示例代码是使用的IBM公司的WebSphere Voice Server SDK（语音服务器软件开发工具包）进行测试的，若要获得该SDK，请访问网址<http://www-4.ibm.com/software/speech/enterprise/ep-12.html>（除了呼叫转移和录音以外的全部功能）。另外，你还可以从VoiceGenie开发人员的工作室站点获得，网址是<http://developer.voicegenie.com/>（包括呼叫转移和录音）。

本书的结构

本书是按照以下几个专题进行编排的：

- VoiceXML的基本概念——本书第1至3章介绍了开始学习VoiceXML时首先需要掌握的有关信息。第1章讲述了VoiceXML的发展历史以及它在因特网应用程序中的使用情况。第2章介绍用于提供电话访问VoiceXML应用程序所用的结构。第3章详细解释了一些可以利用的软件工具和环境。
- VoiceXML用户界面（VUI）——第4至6章介绍了与VoiceXML的用户界面有关的各种问题。第4章介绍用户界面和VoiceXML用户界面的基本概念。第5章从以用户为中心的设计进程的角度来说明VUI的设计。第6章介绍建造原型、评估以及测试应用程序可用性的方法。
- 开发过程——第7章介绍VoiceXML应用程序的开发过程以及需要考虑的有关问题。
- VoiceXML应用程序的开发基础——第8章和第9章介绍着手进行VoiceXML应用程序开发时所需要的基本元素。第8章详细介绍VoiceXML的基本结构、变量和定位技术。第9章介绍与接收用户输入和传递应用程序输出相关的元素和技术。
- VoiceXML的高级开发技术——第10、11、12章以高级VoiceXML应用程序开发为主题。第10章介绍支持访问特定平台的函数、将ECMAScript与VoiceXML相集成、进行语音录制和电话呼叫转接时使用的各种元素。第11章介绍ECMAScript的基本概念。第12章介绍用VoiceXML进行服务器端编程的方法。
- 预测——第13章讲述了随着VoiceXML的持续发展，对某些问题的展望。
- 附录——附录A提供了一个VoiceXML 1.0版的元素列表、每个元素的简要介绍与介绍它的章节。附录B给出了本书开发的完整的在线书店应用程序。附录C简要地介绍了一旦VoiceXML 2.0版推向市场，它对1.0版可能进行的修改。

作者简介

Rick Beasley是朗讯公司的执行经理，他目前管理着几个软件开发项目。他的小组负责开发综合性的因特网/电信服务程序和平台，其中包括Milife Voice Solution这个VoiceXML网关。

在过去的3年里，Rick为移动因特网开发的软件既有网络基础软件也有用户应用程序。他在12年的软件开发工作中开发的产品包括网络管理软件、手持平台和地理信息系统。Rick目前正在研究“蓝牙”（Blue tooth）和基于位置的服务程序。同时，Rick还是与VoiceXML服务程序有关的两项专利的合伙申请人。

Mike Farley是朗讯公司的一名专业技术人员。目前负责从事以用户为中心的软件和信息产品设计。

Mike在无线和有线电信系统及智能网络方面拥有35年的客户服务经验和面向用户的工作经验。他为通过用户为本的设计、方便的交叉功能设计和开发小组、技术培训以及对进程和组织的再设计，为用户提供了人性化的支持。

在过去的两年中，Mike负责为实现VoiceXML、无线标记语言（WML）、Web浏览器和PC用户界面的各式各样的服务应用程序进行以用户为中心的设计。

John O'Reilly是朗讯公司的一名专业技术人员。目前负责领导朗讯公司的VoiceXML平台小组的工作。从1999年3月起，John一方面参与VoiceXML平台小组的工作，一方面又担任VoiceXML应用程序开发小组的主要开发人员。John作为一名专题发言人，在2001冬季因特网之声（VON）上介绍了VoiceXML应用程序的开发问题。John也是一名与VoiceXML相关专利的合伙申请人。

Leon Henry Squire是朗讯公司一名专业技术人员。他领导的开发小组负责开发在增强型媒体资源服务器（eMRS）上运行的TelePortal VoiceXML引擎。Leon在AT&T和朗讯公司中从事了13年的电话技术的开发研究。在软件开发、系统测试和系统设计方面担当技术负责人的角色。他目前是朗讯公司的高级软件开发中心的主要软件开发人员，负责制定开发小组的技术发展方向，该中心位于北卡莱罗纳州的卡里市。

目 录

译者序	
前言	
作者简介	
第1章 VoiceXML的概述	1
1.1 VoiceXML语言	1
1.2 一个VoiceXML应用程序方案	1
1.3 语音浏览	2
1.4 VoiceXML在语音浏览器中是怎样工作的	3
1.4.1 应用程序的输入	3
1.4.2 应用程序的输出	4
1.5 高级网络结构	4
1.6 VoiceXML网关	5
1.7 托管VoiceXML应用程序	6
1.8 选择VoiceXML应用程序	7
1.9 用VoiceXML来开发应用程序	7
1.10 支持的语言和工具	9
第2章 运行VoiceXML应用程序的网络	10
2.1 VoiceXML网关	10
2.1.1 TTS资源	11
2.1.2 ASR资源	12
2.1.3 VoiceXML浏览器和翻译器	13
2.1.4 声音资源软件	14
2.1.5 TCP/IP资源	14
2.1.6 电话资源	14
2.2 电话网络的拓扑	15
2.2.1 有线电话网的拓扑	15
2.2.2 无线电话网的拓扑	17
2.3 小结	19
第3章 开发人员使用的工具	20
3.1 开发环境的类型	20
3.2 可供选择的开发环境和它们的利弊	20
3.2.1 托管的开发环境	22
3.2.2 模拟的开发环境	27
3.3 基于Web的开发环境	28
3.3.1 BeVocal Café	29
3.3.2 HeyAnita FreeSpeech	30
3.3.3 Tellme Studio	31
3.3.4 VoiceGenie 的Developer Workshop	32
3.3.5 Voxeo Community	33
3.4 其他有用的工具	33
3.5 小结	34
第4章 VoiceXML用户界面	35
4.1 用户界面的基本概念	35
4.1.1 用户界面的组成部分	36
4.1.2 CHUI、GUI和WUI	36
4.1.3 Web用户界面	37
4.2 VUI概述	37
4.2.1 VUI的功能	37
4.2.2 VUI的术语	38
4.2.3 VUI的输入/输出	38
4.3 VUI与GUI在设计上的差异	40
4.3.1 Web浏览器用户界面与VUI的比较	40
4.3.2 Web GUI与VUI的比较总结	42
4.4 VUI用户的特点	43
4.4.1 对PC和因特网的使用经验有限	43
4.4.2 处于移动环境之中	44
4.4.3 单一的输入/输出模式	44
4.5 设计VUI时应该考虑的其他重要问题	44
4.5.1 吸引你的目标用户	45
4.5.2 能够代表目标用户群希望执行的任务	45
4.5.3 提供一个适合代码开发的结构	45
4.5.4 尽量减少限制条件造成的影响	46

4.6 小结	46
第5章 VUI设计的概述	47
5.1 软件开发过程的概述	47
5.2 UCD过程的概述	48
5.3 可用性测试	52
5.3.1 VoiceXML的可用性测试	52
5.3.2 测试原则1：尽量降低对用户的记忆和识别过程的要求	53
5.3.3 测试原则2：保持简单而自然的对话框	55
5.3.4 测试原则3：提供清楚的导航和快捷操作方式	56
5.3.5 测试原则4：提供一致的反馈信息和便于使用的错误处理工具	57
5.3.6 测试原则5：保持导航和术语的一致性	59
5.3.7 测试原则6：对选择的声音和声音的个性进行评估	59
5.4 UCD和系统设计的参考资料	61
5.5 小结	61
第6章 VUI应用程序原型和可用性测试	62
6.1 UCD 的前端工作	62
6.2 应用程序的任务分析	64
6.2.1 新技术应用程序开发中的任务分析	64
6.2.2 任务分析的举例	64
6.3 早期的应用程序原型——应用程序流程图	65
6.3.1 建立低保真度的应用程序原型	66
6.3.2 用流程图来说明应用程序的功能	67
6.3.3 根据流程图对应用程序的可用性进行评估	67
6.3.4 应用程序可用性要求举例	68
6.3.5 初始语法设计	69
6.3.6 与开发人员交流信息的结果	71
6.4 VUI脚本——中间阶段的应用程序原型	71
6.4.1 可用性测试举例	71
6.4.2 可重复使用的资源	73
6.4.3 定时问题与资源要求	73
6.5 VoiceXML代码原型	74
6.5.1 可用性测试的前提条件	74
6.5.2 可用性测试过程的概述	74
6.5.3 制定可用性测试方案	75
6.5.4 使用测试结果来改进应用程序的设计	77
6.5.5 完成可用性测试的工作周期	78
6.6 小结	78
第7章 VoiceXML应用程序的开发过程	79
7.1 确定应用程序开发的要求	79
7.1.1 应用程序部署地区使用的口语	79
7.1.2 语法规的定义	80
7.1.3 语音识别与TTS软件工具的选择	80
7.1.4 支持的VoiceXML应用程序技术规范	80
7.1.5 进行数据上的准备	80
7.1.6 内容提供商界面	81
7.1.7 性能、容量和可靠性	81
7.2 确定应用程序的结构	82
7.3 应用程序设计的实现方法	83
7.4 应用程序性能计量参数的定义	83
7.5 VoiceXML应用程序的测试	85
7.6 应用程序的部署	85
7.7 小结	86
第8章 VoiceXML的基本命令	87
8.1 第一个VoiceXML程序	87
8.2 xml元素	87
8.3 注释	87
8.4 vxml元素	88
8.5 form元素	88
8.6 窗体中的项目	88
8.6.1 name属性	88
8.6.2 expr属性	88
8.6.3 cond属性	88
8.6.4 使用说明	89
8.6.5 block元素	89
8.6.6 name属性	89
8.6.7 expr属性	89
8.6.8 cond属性	90

8.6.9 使用说明	90	8.12.6 namelist 属性	104
8.6.10 字段项	90	8.12.7 method 属性	104
8.6.11 filled元素	90	8.12.8 enctype属性	104
8.7 meta元素	90	8.12.9 caching 属性	104
8.7.1 name属性	91	8.12.10 fetchtimeout 属性	104
8.7.2 content属性	91	8.12.11 fetchhint 属性	104
8.7.3 http-equiv属性	91	8.12.12 fetchaudio属性	104
8.8 变量	92	8.12.13 return 元素	105
8.8.1 声明变量	93	8.12.14 程序举例	105
8.8.2 name属性	93	8.12.15 另一个程序例子	106
8.8.3 expr 属性	93	8.13 goto 元素	108
8.8.4 使用说明	93	8.13.1 next属性	108
8.8.5 name属性	94	8.13.2 expr属性	109
8.8.6 expr属性	94	8.13.3 nextitem属性	109
8.8.7 使用说明	94	8.13.4 expritem属性	109
8.8.8 关于变量的其他信息	94	8.13.5 caching属性	109
8.8.9 根文档	95	8.13.6 fetchtimeout 属性	109
8.8.10 会话变量	97	8.13.7 fetchhint 属性	109
8.8.11 影子变量	97	8.13.8 fetchaudio属性	109
8.8.12 变量命名冲突	97	8.13.9 程序举例	109
8.9 if元素	98	8.14 clear元素	111
8.10 if / elseif / else元素	99	8.14.1 namelist属性	111
8.10.1 cond 属性	100	8.14.2 使用说明	111
8.10.2 程序举例	100	8.15 循环	111
8.10.3 嵌套的if/else/elseif 语句	101	8.16 exit元素	112
8.11 param元素	102	8.16.1 expr属性	112
8.11.1 name属性	102	8.16.2 namelist属性	112
8.11.2 expr 属性	102	8.16.3 程序举例	113
8.11.3 value 属性	103	8.17 disconnect元素	113
8.11.4 valuetype属性	103	8.18 小结	114
8.11.5 type属性	103	第9章 VoiceXML的输入和输出命令	115
8.12 subdialog元素	103	9.1 VoiceXML中的输出方法	115
8.12.1 name 属性	103	9.1.1 prompt 元素	116
8.12.2 src属性	103	9.1.2 语音标记	118
8.12.3 expr 属性	103	9.1.3 value元素	122
8.12.4 cond 属性	104	9.1.4 程序举例	122
8.12.5 modal属性	104	9.1.5 audio元素	126

9.1.6 使用举例	126	11.4.2 Date对象	213
9.2 接收用户的输入	131	11.4.3 Math 对象	214
9.2.1 field 元素	131	11.4.4 String对象	215
9.2.2 option元素	140	11.4.5 函数与 Function对象	215
9.2.3 enumerate元素	147	11.5 运算符	216
9.2.4 menu元素	147	11.6 类型转换	218
9.2.5 choice元素	150	11.7 控制程序的执行	218
9.2.6 link元素	152	11.7.1 if/else语句	218
9.2.7 grammar元素	156	11.7.2 case 语句	219
9.2.8 dtmf元素	157	11.7.3 循环	220
9.3 事件	159	11.7.4 with语句	222
9.3.1 事件的类型	159	11.8 小结	222
9.3.2 默认的 catch元素	160	第12章 高级主题和服务器端的应用	
9.3.3 catch元素	161	程序开发	223
9.3.4 throw 元素	168	12.1 语言	223
9.3.5 error元素、help元素、noinput 元素和nomatch元素	171	12.2 创建VoiceXML应用程序	224
9.3.6 reprompt元素	172	12.2.1 Java小服务程序	224
9.3.7 initial 元素	175	12.2.2 PHP脚本程序	229
9.4 小结	178	12.2.3 数据库访问	231
第10章 VoiceXML的高级元素	179	12.3 加载声音文件	239
10.1 record元素	179	12.4 小结	245
10.2 transfer元素	180	第13章 VoiceXML的未来发展前景	246
10.3 script 元素	182	13.1 关于语音合成标记语言的概述	246
10.4 property元素	185	13.2 关于语音识别语法的概述	247
10.5 object元素	196	13.3 多模式应用程序	248
10.6 submit元素	198	13.3.1 多模式应用程序给VUI带来的影响	249
10.7 完成应用程序中VoiceXML部分的开发	201	13.3.2 串行输入/输出	249
10.8 小结	209	13.3.3 非协调式同时输入/输出	250
第11章 VoiceXML中的ECMAScript	210	13.3.4 协调式同时输入/输出	250
11.1 ECMAScript注释	210	13.4 未来用户界面的复杂性	251
11.2 变量的类型	210	13.5 VoiceXML多模应用程序的现状	252
11.3 用户定义的对象	211	13.6 小结	252
11.4 内置对象	212	附录A VoiceXML 1.0的元素	253
11.4.1 Array 对象	213	附录B 在线书店应用程序	255
		附录C VoiceXML 2.0的有关情况	277

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。

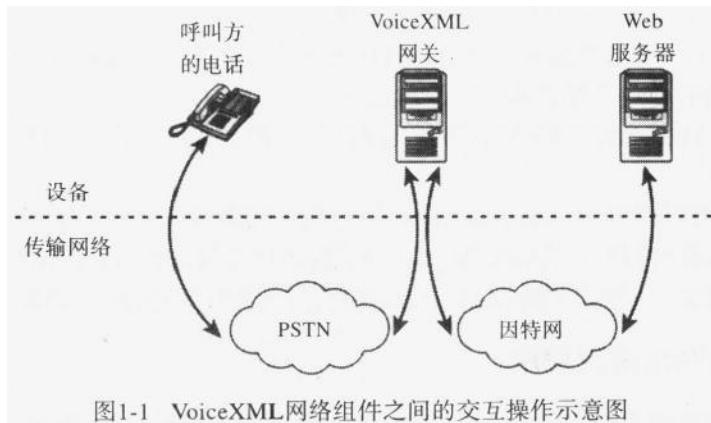


图1-1 VoiceXML网络组件之间的交互操作示意图

- 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账系统。
- 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为了有一个驻留的位置。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声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>
<vxmle>
```

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页的同样的方法，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中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）。

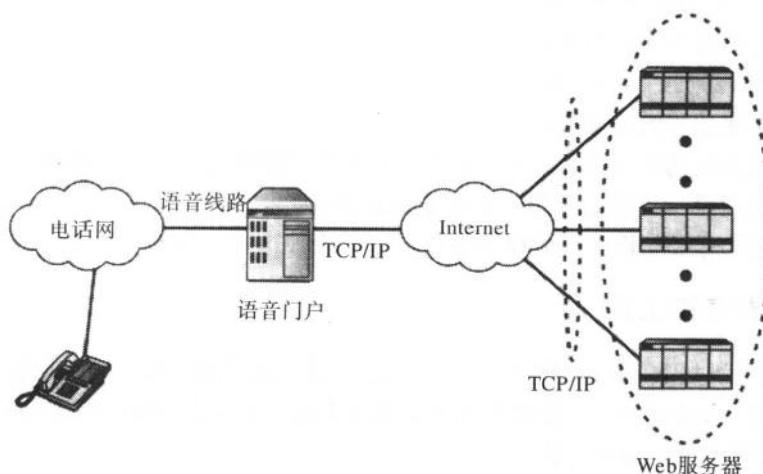


图2-1 运行VoiceXML应用程序的网络的主要组成部分，
它包括语音门户、电话网和因特网

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是有线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，新的电话服务供应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

互操作时用户的感觉。

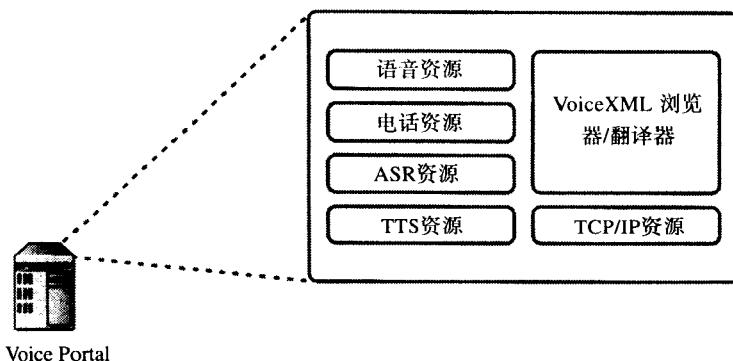


图2-2 语音门户的组成结构

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>
<filled>
```

```

<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阀值的噪声），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如何确保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之间的主干连接。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唯一的电话设备。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。

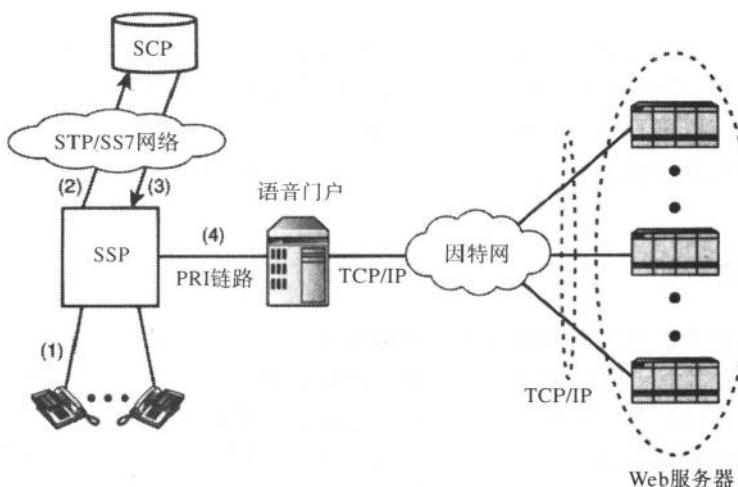


图2-3 触发式有线电话网的拓扑,这种网络也是8xx、
Caller ID等高级智能网服务的基础

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，并且从技术上讲是同时被激活的。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）起着一个数据库的作用，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（SHLR）。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

方使用他的电话。这些数据可以与其他供应商共享，以便在不同网络之间进行“漫游”。

为了减少无线欺诈的机会，因此使用了设备身份识别注册表（EIR）和身份验证中心（AC）数据库。这些数据库分别用于存储覆盖区域内的所有活动无线电话的序列号和植入所有移动电话的密钥。当拨打无线电话时，发送的序列号与密钥相结合，可以防止无线欺诈的产生。

从SS7的角度出发，与STP相结合，它们构成了IS-41无线网络的主要网络组件。图2-4显示了一个无线用户访问与MSC连接的语音门户的情况。

下列步骤描述了一个高层无线拨号触发的电话呼叫的例子，这个例子可以用图2-4来表示。

- 1) 呼叫方拨打电话号码*555-3932。
- 2) 由于MSC已将号码555-3932定为一个被叫号码触发器，因此MSC便根据这个被叫号码向SCP/HLR发送一个查询。
- 3) SCP/HLR对查询的内容进行分析，确定电话应该转发给号码为919-555-8080的语音门户。随后SCP/HLR将路由指令发送给包含语音门户电话号码的MSC。
- 4) 接收到来自SCP/HLR的路由指令后，MSC开始发送一个ISDN用户部分（ISUP）信号序列，这样就可以建立一个到达919-555-8080线路上的门户的ISUP语音链路，将当前的电话转发给语音门户。该链路最终将呼叫方与语音门户直接连接起来。

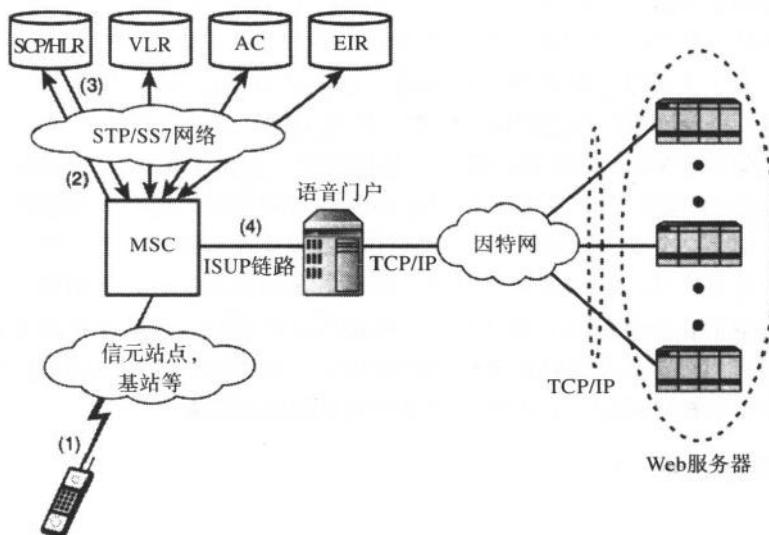


图2-4 用于某些语音门户的无线网络拓扑的高层示意图

- 4) 接收到来自SCP/HLR的路由指令后，MSC开始发送一个ISDN用户部分（ISUP）信号序列，这样就可以建立一个到达919-555-8080线路上的门户的ISUP语音链路，将当前的电话转发给语音门户。该链路最终将呼叫方与语音门户直接连接起来。
- 5) 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。

设计时应该考虑的一些问题

当你建立无线电话网拓扑时，请牢记以下设计方面的注意事项：

- 就像有线电话服务一样，你应该注意你所部署的VoiceXML服务与现有的电话用户服务之间的交互操作问题。
- ISUP协议标准不支持2B信道转接的功能或任何与其类似的功能。因此，当ISUP用作语音门户与MSC之间的协议时，所有的电话传输将要占用语音门户上两个语音通道，你将会遇

到前面介绍的有线电话网一样的传输问题。

- 在这种情况下，转发电话的计费将会变得稍微复杂一些，因为原始的电话将被当作无线电话来收费，但是出局的转发电话很可能按有线电话来计费。因此，必须对这两种收费方式进行协调。

2.3 小结

本章介绍了今天的许多语音门户结构的主要组件。本章还讲述了你在着手进行应用程序的设计时应该了解的一些基本注意事项。在后面各章中将要进一步阐述应用程序设计方面的问题，并且还要深入介绍VoiceXML应用程序的实际开发。

第3章 开发人员使用的工具

用VoiceXML进行应用程序开发时，你的感觉与大多数Web开发人员的感觉是相似的。感觉最不同的时候是你着手进行VoiceXML应用程序开发的时候，也就是你建立应用程序开发环境的时候。值得高兴的是，你可以使用许多很有帮助的开发工具。

本章将要介绍如何选择这些开发工具，以及如何将这些工具组装成一个有效的开发环境，以满足你的特定需要。我们将要讲述各种编码和调试工具，还要介绍一些商用工具箱和包含许多必要工具的基于Web的开发环境。

本章假设大多数开发人员在对VoiceXML技术进行分析以后，有兴趣开始进行方便的VoiceXML应用程序开发工作。

当你学习一项新技术时，没有什么比你所用的工具出现问题更令人头疼的了。因此你选择的开发环境必须尽可能做到便于安装、配置和使用。为此，我们将要推荐一些适合一般情况使用的工具，并且要说明如何购买、安装和配置这些工具。我们将设法使工具的安装尽可能简单明了，并且提供了几个替代安装方法，供你在遇到问题时灵活使用。

随着你的经验的积累，你可以选择不同的工具来满足你的新要求或个人喜好。这样我们就可以进入本章的最后一项内容的介绍，即对我们可以得到的某些工具作一个总结。VoiceXML是一种较新的语言，并且与其他新技术一样，它的前景是不断变化的。但是，了解某些工具和供应商的情况，将有助于你始终处于这个领域中的最新开发技术的领先位置。

3.1 开发环境的类型

你的开发环境可以是你自己建立的一个单独的VoiceXML模拟网络，也可以是你经常免费访问的用于开发和测试的基于Web的工具。这些基于Web的工具功能非常强大，使用比较容易，并且它的功能非常广泛。对于大多数编程人员来说，使用这种工具是最好的。

在下面各节内容介绍的开发环境中，你可以继续使用你已经熟悉的各种工具，比如你的文本编辑器和Web服务器。通过了解这种开发环境是怎样组合在一起的，你就能够更好地理解VoiceXML网络是如何工作的。

前面我们讲过，许多开发人员可能更喜欢使用基于Web的工具，因此下面我们很快将对它们进行介绍。但我们首先要介绍一种利用你现有的编程工具并且能够使语音Web更容易理解的开发环境。为什么要首先介绍这种开发环境呢？因为，有些基于Web的工具的功能已经变得如此广泛，以至于它们隐藏了编程人员应该了解的VoiceXML重要元素。

我们首先要介绍使用不同工具的利弊。

3.2 可供选择的开发环境和它们的利弊

当你选择开发环境时，你必须在你自己的模拟开发环境与使用基于Web的工具之间权衡利弊。

如果使用你自己建立并运行的开发环境，你将会得到最大的控制权，但是这样做的复杂性超出了某些读者希望的程度。相反，许多因特网工具已经能够支持大多数开发人员想要的基本功能，不过这些工具究竟如何运行，则是你无法控制的。

当我们谈论工具箱时，我们实际上是在谈论若干不同的事情。这里的“工具箱”包括许多单一供应商的解决方案，它们可能是集成式开发环境（IDE），也可以是一套松散集成的个人开发工具，或者是基于Web的开发环境，一种比较新的工具类型。

下面我要介绍3种基本的开发环境：

- 托管的开发环境（基于PC的托管）。
- 基于Web的开发环境。
- 模拟器开发环境（基于PC的模拟）。

在托管的开发环境中，你使用你最喜欢的文本编辑器进行编码，再将你的VoiceXML代码放在一个Web服务器上，并且通过语音服务提供商（VSP）来访问你的应用程序。这是VSP的主要功能，因此所有VSP都支持这个功能，有些VSP还免费提供开发人员程序。

使用VSP而不使用你通过PC上的麦克风和扬声器（或头带耳机）与之进行交互操作的模拟器，你会得到许多好处。暂时忽略VSP所拥有的任何基于Web的开发工具，通过使用VSP，你将能够使用“真正”的语音Web界面——电话。通过使用与你的用户相同的界面，你能够很容易确定存在的问题，例如，界面问题和网络延迟。

但是有若干个VSP已经走到了前面，它们不仅能够提供语音服务，而且也能够提供基于Web的各种工具。它们包括许多基于Web的开发环境，能够提供你编写应用程序所需的一切工具。有些VSP甚至通过托管你的VoiceXML而不需要将你的VoiceXML放到Web服务器上去，同时，如果你希望的话（例如，如果你要编写服务器端的代码的话），你可以链接到一个远程服务器，这就要求你使用托管的开发环境。

由于基于Web的开发工具非常容易安装和配置，因此我们可以通过这些公司的开发人员程序来获得这些工具。我们将其称为基于Web的开发环境。

如果你只对高层VoiceXML应用程序的开发感兴趣，而对某些基础技术没有兴趣，那么你可以跳过对模拟器、XML编辑器和诸如此类的问题的学习，你只需要进入某个开发人员程序，你就能够在几分钟内学会编写和测试VoiceXML代码。无论你决定采用何种方法，你都可以访问模拟器和VSP的Web站点，并且了解它们正在做什么。如果你决定使用它的完整开发环境，每个供应商都能够提供很有价值的开发工具。

那么为什么不设法申请某些基于Web的工具箱并且用它们进行应用程序的开发呢？在许多情况下，这是最佳方案，不过有些情况下需要组建一个定制的开发环境。

如果你采用拨号接入因特网的方法并且只拥有一条电话线，那么你可能不喜欢使用基于Web的开发环境。如果那条电话线也用于接入因特网，你必须切断你的因特网会话，拨打电话号码来测试你的应用程序，然后重新建立一个与因特网的连接。

有些VSP通过“检验器”来解决这个问题，该检验器负责检查你的VoiceXML的语法和文本界面，使用文本界面，你可以通过键盘输入与你的应用程序进行交互操作。这些情况都是很有用的，不过这些问题已经远远偏离了你的应用程序如何运行的问题。

使用模拟器的另一个好处是，通过安装、配置和维护模拟器，你可以更多地了解VoiceXML翻译器和网关内部是怎样工作的。如果你对此感兴趣，你可以采用模拟器方法。

下面几节内容将要详细介绍每种开发环境的情况。

3.2.1 托管的开发环境

从高层次上说，VoiceXML项目的实现可以包括编写VoiceXML代码、提供诸如声音文件和语法等支持数据、对代码进行调试，以及通过VoiceXML网关来运行应用程序等。为了完成这些任务，起码需要的一套工具应包括，一个文本编辑器、对Web服务器的访问工具，以及VSP支持的VoiceXML网关。下面我们将要介绍每一种必要的组件和你可以从中进行选择的一些选项。

编辑器

在开始编写VoiceXML代码时，真正需要的惟一工具是文本编辑器。在Windows中，你可以使用记事本（Notepad）和写字板（Wordpad）（请记住使用“文本文档”这个文件类型）。你还可以使用许多出色的共享软件和免费软件的程序员文本编辑器，它们配有高级文件管理和文本颜色编码等特性。

VoiceXML编辑工具也逐渐成为可以利用的工具，虽然允许你输入DTD的任何XML编辑器都可以使用，但是现在也有几种特定的VoiceXML编辑器可以利用，比如Voxeo和HeyAnita公司提供的几种产品。Tellme网络也提供了一种可供下载的EMACS VoiceXML主要编辑器模式。

Voxeo公司的Designer是专门用于VoiceXML应用程序开发的一种编辑器。使用Designer，你可以通过流程图视图或实际的代码来查看和编辑你的程序。虽然像这样的工具并不是完美无缺的，但当你开始进行VoiceXML编程时，你会对它们感兴趣并且它们可能对你很有帮助。

Web服务器

当你建立了一个VoiceXML程序之后，在托管开发环境中进行的下一个操作步骤是将你的程序放到一个适当的Web服务器上去，这样你就可以将你的应用程序的URL链接到VoiceXML网关控制的一个电话号码。虽然有些VSP允许将你的VoiceXML代码上载到他们的服务器中，但是这样做会限制你编写服务器端代码的能力。

```
The changes required to your Web server vary from VSP to VSP and, depending on
➥your service provider and Web server, no additional MIME types may be
➥necessary. Browse your VSP's FAQ to learn its current requirements for the
➥particular Web server software you are using. Following is an example of the
➥kind of change you may have to make to add a new MIME type to your Web
➥server:  
server:File extension: text/vxml  
Content type: VoiceXML source code  
MIME type: text/xml
```

你可能在不同平台上看到的其他MIME类型是应用程序/x-vxml应用程序/vxml和文本/x-vxml。另外，某些VSP可以要求使用其他的MIME类型，例如支持的声音文件格式，因此请务必弄清楚你的VSP的要求。

如果你使用Apache HTTP Server，那么请修改httpd.conf文件，以便添加新的.MIME类型。

如果你使用微软公司IIS Server，那么请打开Management Console并选定Action，在Action菜单中选定Properties，File Types，这时你就可以输入新的MIME类型。

如果你使用Java Web Server，那么请在你的Web服务器中打开Management Console，选定Management link，然后选定MIME Types，在这里添加新的MIME类型。

运行你的VoiceXML代码

当你保存了一个简单的VoiceXML程序并把它放到你的Web服务器上后，接下来操作步骤是向VSP申请一个账户。我们要介绍的服务提供商所做的事情远远不只是提供VSP服务。实际上，VSP服务根本就不是他们的重点任务。不过这些公司确实通过开发人员程序来提供VSP服务，这些开发人员程序对它的成员来说是开放的。

我们将要介绍的一些开发人员程序包括：BeVocal (www.bevocal.com)、HeyAnita (www.heyanita.com)、Tellme Networks (www.tellme.com)、VoiceGenie (www.voicegenie.com)和Voxeo (www.voxeo.com)。在撰写本书的时候，这些公司为进入它们的开发人员程序的编程人员提供了免费账户，有些公司还提供免费长途电话。

只要VSP提供免费开发人员程序，那么你就有必要尽可能多地熟悉这些程序，直到你开发出一个满意的首选程序为止，也许以后你还要始终关注它的新特性。

若要运行你的VoiceXML应用程序，你必须将该应用程序的URL链接到你选择的VSP所提供的一个电话号码（并且可能是ID与PIN的组合）。当你进入一个VSP的开发人员程序以后，请使用下面的操作过程将你的应用程序链接到VSP提供的一个电话号码。

BeVocal Café

使用你的用户ID和口令登录到BeVocal 的Café开发人员网络。登录以后，你会看到File Management（文件管理）页。接着进入Remote URL Based Application Development（基于远程URL的应用程序开发）这部分。输入你的应用程序的URL（见图3-1）并单击Add按钮。这时URL便出现在页面底部你的URL的列表中。这里你可以单击Activate，使你的应用程序开始运行。

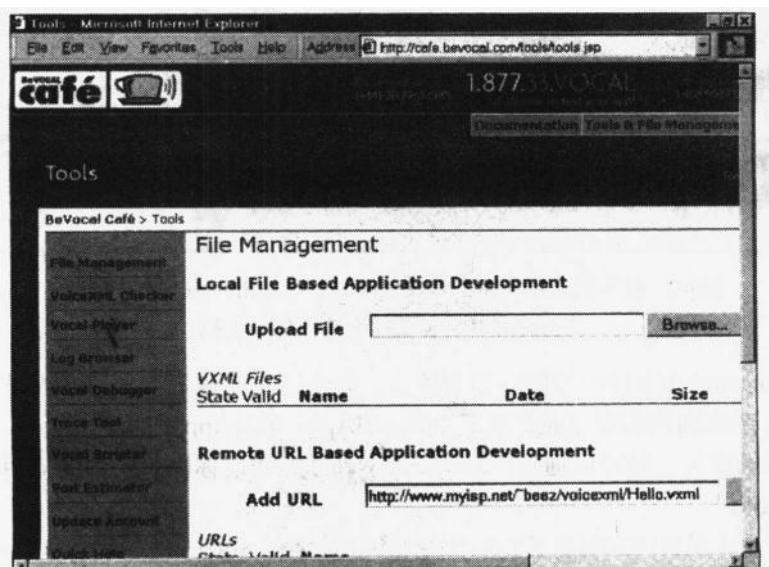


图3-1 将你的应用程序的URL输入Cafe的File Management屏幕，以便使用BeVocal来运行你的代码

当你调用BeVocal Café来访问你的应用程序时，系统提示你输入你的PIN和用户ID。然后你就可以立即运行你的应用程序。

HeyAnite Freespeech

使用你的开发人员ID和PIN登录到HeyAnite Freespeech 开发人员网络，选定URLApp 选项卡，在这里输入你的应用程序的URL（见图3-2），然后，你可以单击Check Syntax（检查语法），以便确认FreeSpeech正确地访问了你的VoiceXML代码，单击Update使你的应用程序开始运行。

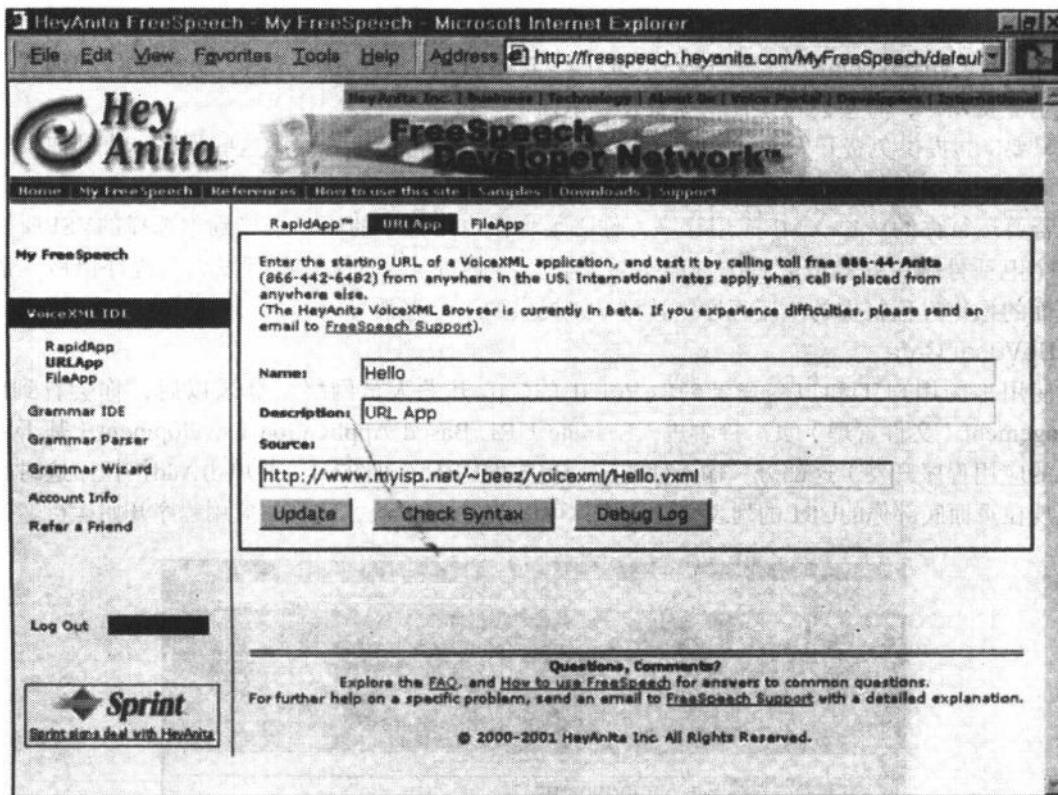


图3-2 将你的应用程序的URL输入Freespeech的URLApp选项卡，以便使用Heyanita来运行你的代码

当你调用Freespeech来访问你的应用程序时，系统提示你输入你的开发人员ID 和PIN。在你完成登录以后，系统询问你究竟是想要访问RapidApp、FileApp还是URLApp。对URLApp的菜单选项作出相应的应答，就可以通过Freespeech Web站点访问你以前配置的应用程序。

Tellme Studio

使用你的开发人员ID和PIN登录到Tellme Studio，在Application URL选项卡上，输入你的应用程序的URL（见图3-3），单击Update，通过拨打访问电话号码并且在系统提示你输入开发人员ID和PIN时，你就可以运行你应用程序。

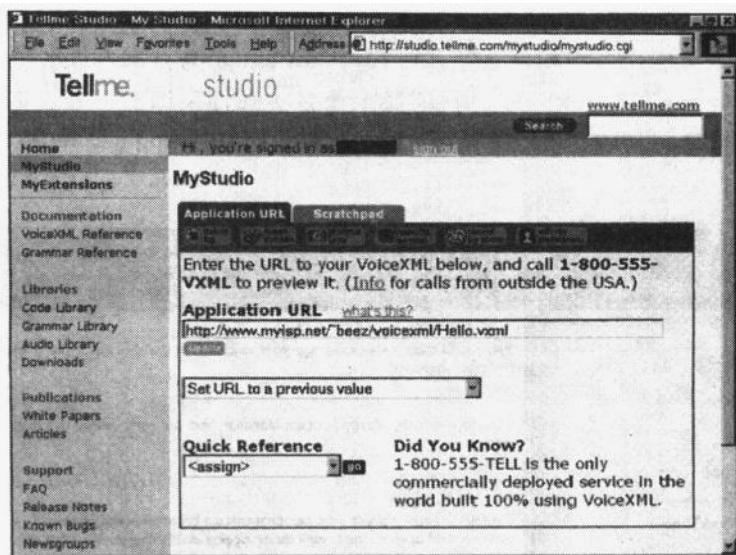


图3-3 将你的应用程序的URL输入Studio的Application URL选项卡，以便使用Tellme来运行你的代码

VoiceGenie Developer Workshop

使用你的用户名和口令登录到Voice Genie的Developer Workshop。在main workshop（主工作室）页面上，从Tool类别中选定Manage Extensions，接着在Manage Extensions屏幕上输入你应用程序的URL（见图3-4）并单击Add Extension按钮，下一屏将显示你的应用程序分配到的分机号和访问电话号码。

你可以通过拨打访问电话号码并且在提示你输入你被赋予的分机号时运行你的应用程序。

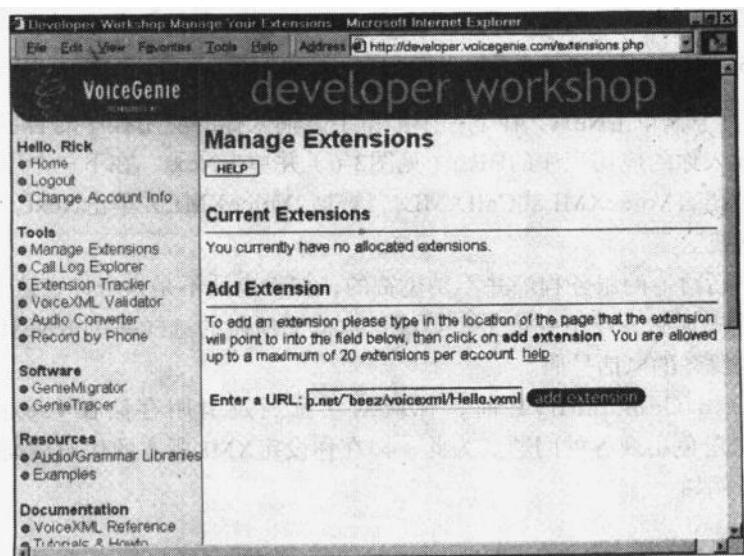


图3-4 将你的应用程序的URL输入Developer Workshop的Manage Extensions屏幕，以便使用Voice Genie来运行你的代码

Voxeo Community

使用你的开发人员用户名和口令登录到Voxeo Community开发人员网络。当你登录以后，Voxeo Account Manager（Voxeo账户管理器）便显示在你桌面上，在左面的显示框中选定Applications选项下的<add new app>（见图3-5）。这样就打开了Application Provisioning Wizard（应用程序设置向导）。

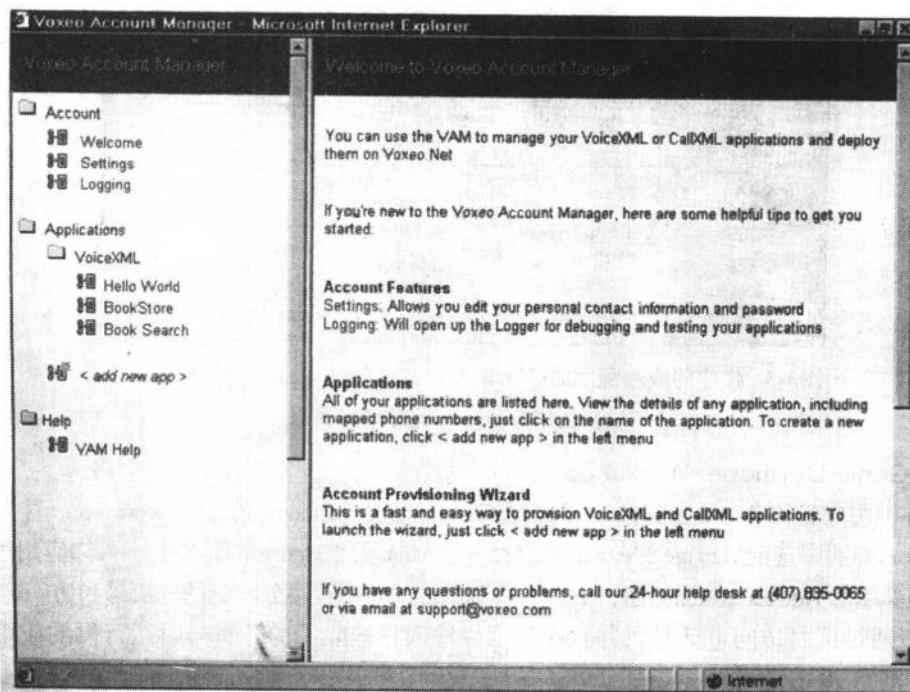


图3-5 通过选定<add new app>来打开Voxeo 的Application Provisioning Wizard

在向导的第1页上，单击Next，在下一个屏幕上，输入你的应用程序的名字并单击Next，在下一个屏幕上，输入你的应用程序的URL（见图3-6）并单击Next，在下一个屏幕上，系统将提示你选择你的标记语言VoiceXML或CallXML，请选定VoiceXML并单击Next，然后系统提示你选择一个电话交换机。

请注意，打电话给你的服务程序并不是免费的，如果电话不是来自你在这里选定的交换机，那么就要收取长途电话费。在下一屏上，请确认你前面所做的选择并单击Next。最后一个屏幕上列出了访问你的服务的电话号码。

在你调用Voxeo Community之前，你应当注意通过引用存储在Voxeo上的Nuance的VoiceXML DTD来避免出现各种问题。为此，请在你设定XML版本的代码后面的这一行代码中设定文件的类型。例如：

```
<?xml version="1.0"?>
<!DOCTYPE vxml PUBLIC '-//Nuance/DTD VoiceXML 1.0b//EN'
  'http://community.oxeo.com/vxml/nuancevoicexml.dtd'>
<vxml version="1.0">
```

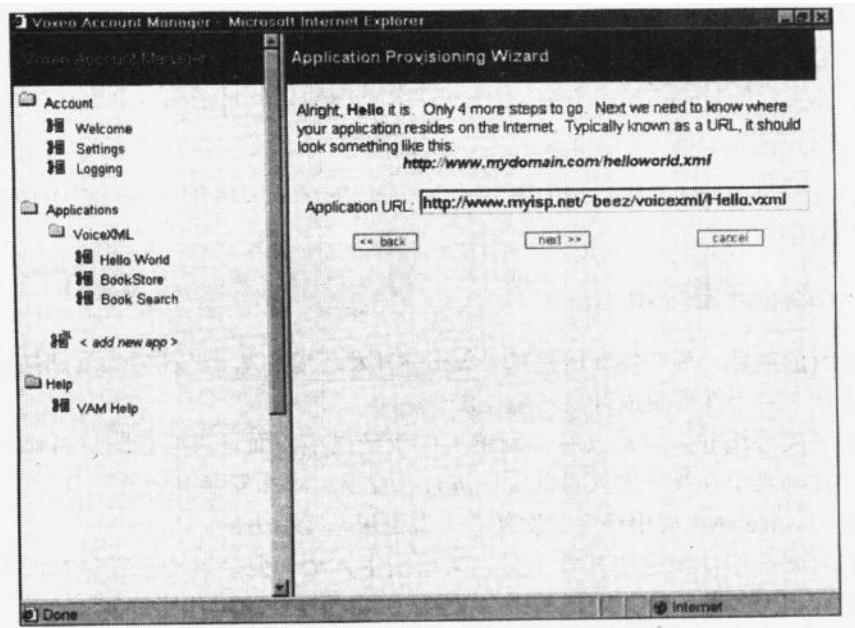


图3-6 将你的应用程序的URL输入向导的Application URL文本字段

3.2.2 模拟的开发环境

前面我们讲过，有时你可能希望或想要使用模拟器来运行你的VoiceXML代码。IBM公司的WebSphere VoiceServer SDK就是用于这个目的的一种产品。

模拟器的运行需要有一个高质量的麦克风和扬声器或者将麦克风和扬声器结合在一起的头戴耳机。一般来说，头戴耳机非常适合用于语音识别，而具备某种消除噪声功能的麦克风也很适合进行语音识别。通常而言，如果你的麦克风无法使用Windows 的Sound Recorder实用程序进行高质量的录音，那么当它与桌面VoiceXML模拟器结合起来使用的时候，它所产生的结果将会让你失望。

下面是分步骤来安装和配置该产品的具体说明，以及如何使用它来运行你的VoiceXML代码的说明。

IBM 公司的WebSphere Voice Server SDK

本产品可以从IBM公司的alphaWorks站点（网址是alphaworks.ibm.com）下载，也可以通过CD 进行下载，该CD可以向alphaWorks站点订购。如果要下载该产品，你必须拥有200MB的磁盘空间（如果你是从Web下载，则需要330MB的磁盘空间），配有Service Pack 6a的Windows NT 4.0或更新版本，太阳公司的Java Runtime Environment (JRE) 1.3版（它包含在分销的VoiceXML SDK中），以及Adobe Acrobat Reader 4.0.5版（也包含在分销的VoiceXML SDK中）。在安装该产品的过程中，如果你的PC没有配备正确的JRE，那么系统将会向你作出提示。

当你完成各个安装步骤并且你的系统已经重新引导之后，请使用Audio Setup程序配置你的麦克风。若要测试该产品的安装情况，请运行包含的示例程序，方法是在Windows桌面上依次

选定Start, Programs, IBM Websphere Voice SDK, Audio Sample。该命令应该能够启动IBM的示例程序开始运行，它将录制一条消息，并且播放该消息。

若要运行你自己的VoiceXML程序，请从包含你的源代码的目录输入下面这个命令：

```
%IBMVSD:\bin\vsaudio_en_GB HelloWorld.vxml
```

然后你就可以与你的HelloWorld程序讲话。Voice Server SDK配有示例程序和文档，包括一个出色的程序员指南。

3.3 基于Web的开发环境

我们要介绍的最后一一种开发环境是基于Web的开发环境。基于Web的工具和站点介绍起来有些困难，它类似于一个在线集成开发环境（IDE）。

由于这种开发环境非常容易安装，配置工作量又很小，而且目前它的价格比较低（大多数情况下，如果你向某个在线开发人员的社区进行了注册，那么安装这种开发环境是免费的），因此它对我们这些VoiceXML应用程序开发新手来说是很有吸引力的。

虽然我们将要介绍使用这些开发工具的某些基本方法，但是这些工具离我们非常遥远，而且变化的可能性很大。惟一的正式版本是在Web上，它的变化难以预测。

由于基于Web的开发工具便于安装并且价格便宜，因此很难在竞争中击败它们。如果你选用这种工具，那么你将会比使用其他方法更快地进入VoiceXML应用程序的开发工作去，你只需要选用一种开发人员程序，几分钟内你就可以编写和运行VoiceXML应用程序了。

除了本章开头介绍的将电话号码映射到应用程序和运行VoiceXML代码时使用的必备工具以外，基于Web的VoiceXML工具箱还包括下列工具的一部分或者全部：

- 代码检验器——本工具用于检查你的VoiceXML代码的语法并报告代码中存在的错误。它可以是一个较大的文本编辑器的一部分，也可以是你将代码粘贴到的一个独立工具，以便对代码进行快速检查。
- 日志查看器——使你能够查看在VSP的VoiceXML网关上运行的你的代码产生的日志。这个日志包含了有关你的应用程序的详细信息，包括错误、变量的计算、程序的控制流程、HTTP首标数据、语音识别状态和其他的一些信息。当应用程序运行或者存储和从前面运行的应用程序检索时，可以对日志进行动态查看。
- 文本界面——用于提供一个文本界面，以取代实际的语音界面。使用文本界面后，当你的应用程序提示你输入有关的信息时，你就可以通过键盘上输入你要输入的信息。虽然你的实际用户可能早就不使用这种界面了，但是对于某些调试操作来说，这个界面是非常有用的，并且在通过拨号调制解调器来使用基于Web的工具箱时，文本界面几乎是必不可少的。
- 支持工具——通过连接一个VSP的开发人员程序，你可以使用它提供的有关技术支持，包括如何使用工具箱本身的文档。此外，它还提供了诸如VoiceXML技术规范之类的文档，其编排格式比单个大型PDF文档要好。最重要的是它提供了另外的许多文档，用于详细描述支持的VoiceXML元素和属性、VSP喜欢采用什么样的语音文件格式、编程人员的常见问题（FAQ）、新闻组、教程和其他许多非常有用的信息。

本章的下一节将要介绍你可以得到的许多不同种类的基于Web的工具箱的某些功能。由于他们是作为VSP来运行的，因此我们将要观察一些我们已经熟悉的某些站点。这些基于Web的开发环境包含了各种各样的开发工具，但是在分别介绍每个VSP的开发环境之前，我们首先要讲述一些通用的开发工具。

3.3.1 BeVocal Café

你可以在网址cafe.bevocal.com上找到Be Vocal Café。本章的前面部分中显示的它的File Management界面，除了可以用于链接到你的应用程序URL外，也支持将你的应用程序上载到它的Web服务器上。

另外，Café拥有一个称为VoiceXML Checker的代码检验器，称为Log Browser的日志浏览器，称为Vocal Scripter的文本界面和支持工具。你可以从屏幕左边的导航菜单中选择你需要的各个工具，工具本身位于主窗口中。

Café的代码检验器VoiceXML Checker能够快速报告错误所在的代码行的号码。

日志查看器Log Browser用于报告来自许多来源的跟踪消息，这些消息源包括：

- 一般流程跟踪消息。
- 定制消息。
- 事件跟踪消息。
- 分析消息。
- 字段填充跟踪消息。
- 平台消息。
- 变量跟踪消息。
- 识别信息。
- 出错消息。
- HTTP首标信息。

日志颜色代码用于根据以前的类别来跟踪各种消息，以便在调试时能够更加容易地对日志进行扫描。你可以通过选择你希望看到的跟踪消息的类型，以及取消选择你不感兴趣的的消息类型，对日志项目进行筛选。

Café的文本界面称为Vocal Scripter。使用Vocal Scripter，你除了可以用文本界面与你的应用程序进行交互操作外，Vocal Scripter还有一个令人感兴趣的功能，那就是你可以用“批量”方式进行操作，它提供的一个文件包含了在某种情况下对你的应用程序的提示可以做出的多个应答。这是用于递归测试的一个非常出色的特性，并且它使其他类型的测试变得不是那么单调。

除了上面已经介绍的几种工具外，BeVocal还提供了一个Port Estimator(端口估算器)，用于确定支持你的应用程序所需的端口数量。你告诉估算器如何使用你的应用程序，例如呼叫到达的速率和电话持续的时间，估算器则告诉你在某个端口范围内有多少个呼叫者将被拒绝。

Café包含了良好的支持特性，包括文档、新闻组、常见问题（FAQ）、链接和发布通知。

3.3.2 HeyAnita FreeSpeech

前面我们介绍过如何使用HeyAnita的FreeSpeech VoiceXML IDE（可从freespeech.heyanita.com网址下载）来链接你的应用程序的URL和运行你的VoiceXML代码。

IDE也支持RapidApp编码方法，使用这种编码方法时，你将你的代码输入文本显示屏，该代码将被存放在HeyAnita的Web服务器上（见图3-7）。在屏幕中输入代码并单击Update按钮，你就可以立刻测试你的代码。这对于学习和尝试单文档应用程序是很有帮助的。

你也可以使用FileApp选项卡将多页文档上载到HeyAnita的服务器。这样你就可以上载你全部的VoiceXML文件和支持文件，例如，录制的声音文件和语法，而不是在你自己的Web服务器上托管这些文件。

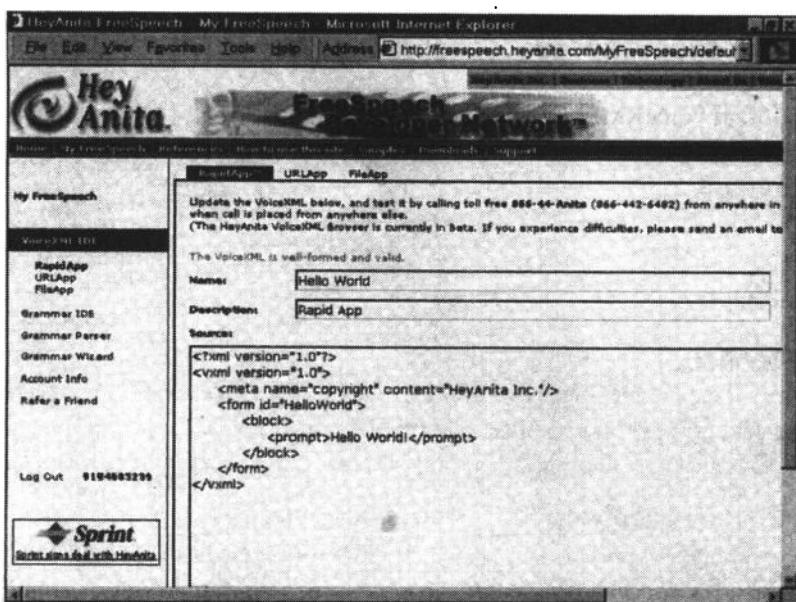


图3-7 登录后，HeyAnita的FreeSpeech就显示RapidApp选项卡上默认的VoiceXML IDE

HeyAnita的代码检验器将被纳入VoiceXML IDE中你可以将网关直接放入VoiceXML代码的所有位置。当你更新RapidApp屏幕中的代码时，当你选定文件以便将它上载到HeyAnita服务器中去时，以及当你在URLApp选项卡中输入你的应用程序的URL的时候，你可以通过单击Check Syntax按钮来运行该检验器。

若要访问HeyAnita的日志查看器，请在RapidApp、URLApp或FileApp选项卡中单击Debug Log按钮。

HeyAnita目前不支持文本界面。

除了我们已经介绍的几种标准工具外，FreeSpeech还包含两个非常有用的语法工具，即Grammar IDE（语法集成式开发环境）和Grammar Wizard（语法向导）。Grammar Wizard简化了编写语法的任务，特别是当你开始使用它们进行工作时，它将向你显示几个输入屏幕，并且创建一个语法。你可以输入语法的名字和它的作用，然后输入各个语法项目。当你创建了一两个

语法并且逐步对这项工作熟悉之后，你就可以使用编辑器来更快地创建语法。

Grammer IDE可以用于为较大的应用程序编写语法。使用这个工具，你可以检查语法的文法，而不会增加应用程序的开销。

3.3.3 Tellme Studio

Tellme Studio可以从studio.tellme.com网址下载。除了它的Application URL选项卡可以用于链接你的应用程序外，Studio还配有一个Scratchpad选项卡（见图3-8），用于运行单页VoiceXML文档。这里你可以直接将VoiceXML代码输入该编辑器，单击Update按钮，你就可以立即调用Tellme来运行代码。

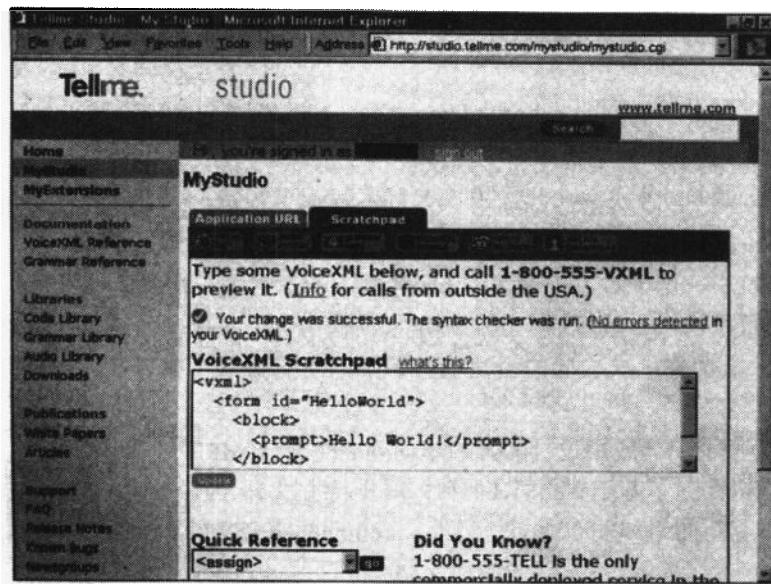


图3-8 Tellme的MyStudio可以打开Scratchpad选项卡，
你可以运行单页VoiceXML文档

Studio的代码检验器、文本界面、日志查看器和其他工具都集成在Scratchpad和Application URL选项卡中。打开这些工具屏幕后，你就可以通过各个选项卡下面的一排按钮来调用这些工具。这里可访问的工具有：

- Debug Log (日志查看器)。
- Check Syntex (代码检验器)。
- Grammer Tools。
- VoiceXML Terminal (文本界面)。
- Record By Phone。

Tellme Studio 的Debug Log能够显示下面5种主要类型的跟踪消息：

- 一般流程跟踪消息。
- 事件跟踪消息。
- 字段填充跟踪消息。

- 变量跟踪消息。
- 错误和定制消息。

日志消息是用颜色编码的消息并且可以按类型来进行筛选。

代码检验器的详细检查结果可以通过单击Check Syntax按钮来进行访问，它只是用来检查你的代码并报告出现的错误和错误所在的代码行号。当你更新Scratchpad或Application URL屏幕时，检验器就会被调用。

使用VoiceXML Terminal，你可以与你的应用程序进行交互操作，并且使用文本输入对终端上的文本提示作出应答。

除了基于Web的标准特性外，Studio还配有Record By Phone 和Grammer Tools等特性。使用Record By Phone，你可以呼叫 Tellme并录制一个消息。然后便以电子邮件的形式将声音文件作为附件发送给你。

Grammer Tools包括了几种用于建立语法的有用工具。Grammer Tools IDE类似于VoiceXML界面，它配有一些特定语法的菜单按钮，用于：

- Scratchpad——这是默然语法选项卡，在这个屏幕上你可以将语法输入该文本窗口，然后单击Check Grammer按钮，以检验你输入的语法代码。
- Grammer URL——用于对位于你的Web服务器上的语法进行检验。
- Parse——你可以输入一些短语，以便确定如何使用你上次检验的语法来识别这些短语。
- Generate——显示你上次检验的语法能够识别的所有短语，或者显示一个可识别短语的随即举例。
- DTMF——为一个单词或用逗号隔开的单词列表生成DTMF序列。

Tellme的支持特性除了基于Web的IDE和标准支持方式外，还包括可重复使用的源代码模块和文档，例如FAQ、新闻组和发布通知。另外，Tellme的声音文件和语法库也是非常有用的。

3.3.4 VoiceGenie 的Developer Workshop

VoiceGenie的Developer Workshop可以从developer.voicegenie.com网址下载。VoiceGenie的Developer Workshop的欢迎屏对左窗格中的主导航屏（见图3-9）进行了介绍。该主窗格给出了高层描述，并且可以链接到开发环境、支持特性及开发商资源。

前面我们介绍过如何使用VoiceGenie 的Developer Workshop中的Manage Extensions屏幕来建立你的应用程序的URL。

Developer Workshop的代码检验器称为VoiceXML Validator。该检验器并没有配备文本输入窗口，但是你既可以输入文件的URL，也可以在你的PC机上选择一个要检验的文件。该检验器不仅能够检验你的代码，而且可以用流程图的形式直观地显示你的应用程序。

Developer Workshop的日志查看器是Call Log Explorer。使用这个工具，你可以按时间顺序浏览以前的呼叫日志，并且可以访问关于某个日志的详细信息。

VoiceGenie的文本界面采用了不同的运行方式，GenieTracer是个基于文本的VoiceXML翻译器，用于以本机或者非在线方式来开发你的应用程序。它既可以接受URL，也可以接受本地文件，并且允许你用批处理方式和交互方式与应用程序进行交互操作。GenieTracer可以从

Developer Workshop站点下载。该工具目前还是个测试版，但是对于用非在线方式模拟应用程序的用户来说，它的应用前景十分看好。

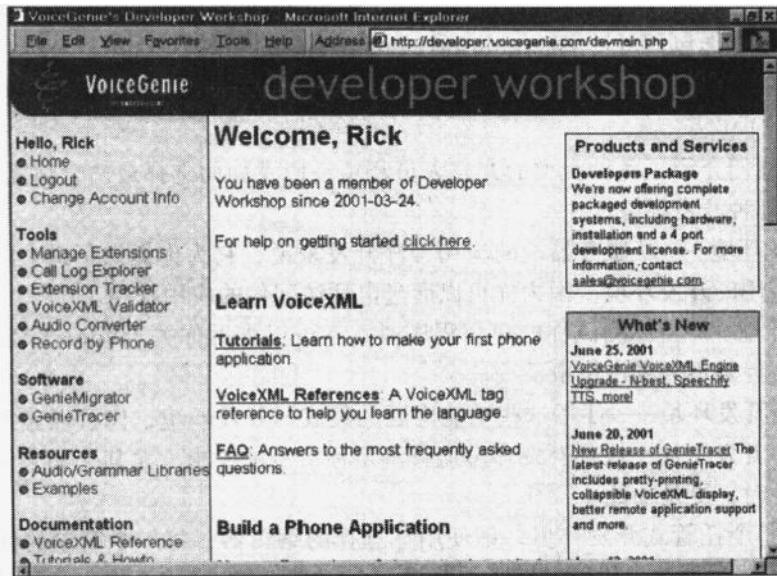


图3-9 VoiceGenie的Developer Workshop打开时配有对主要工具和资源的描述，并且有一个标准导航菜单

除了你想要的开发工具外，VoiceGenie的Developer Workshop还提供了Audio Converter（声音文件转换器）和Record By Phone（电话录制声音文件）这两个工具。Audio Converter能够将你声音文件转换成VoiceGenie的平台所支持的Dialogic Vox格式。通过调用Record by Phone工具，你可以录制你的声音文件，它通过电子邮件将用Vox格式录制的最终声音文件发送给你。

Developer Workshop配有一系列支持特性和资源，它们包括示例代码、教程、白皮书、开发者论坛、链接和FAQ。

3.3.5 Voxeo Community

前面我们按照各个步骤介绍了如何使用Account Manager（可从community.voxeo.com网址下载）在Voxeo上建立应用程序的方法。VoxeoIDE不支持向它的Web服务器上载你的代码。

通过进入Account菜单，你就可以访问Voxeo的日志查看器。该日志能够显示各种跟踪消息。Voxeo Community不提供任何文本界面。

支持工具和文档均位于Voxeo Community站点上。可以访问的资源包括教程、示例代码、新闻组和一个录制好的声音文件库。

3.4 其他有用的工具

由于大多数应用程序都要与录制好的声音文件打交道，并且你需要进行文件格式的转换和

抽样，因此声音编辑器或转换器将对你大有帮助。

这些工具中许多都是现成可以得到的，至少包括两个高质量的共享软件包。Voxtudio是一个专业电话声音软件包，可作为共享软件加以利用，可以从www.xentec.be网址下载。Goldwave是另一个很好的共享软件包，可从www.goldwave.com/release.html网址下载。

3.5 小结

在本章中，我们介绍了各种类型的开发人员环境，以及如何选择最适合你需要的开发环境，我们讲述了下列几种开发环境：

- 托管式开发环境——从服务器端的应用程序开发来说，它为你提供了最大的灵活性，它是一种非常实用的开发环境，因为你可以通过电话访问你的应用程序。
- 模拟开发环境——当你测试你的应用程序时，它可以告诉你关于*VoiceXML*网关的组件方面的知识，但是不需要占用电话线。
- 基于Web的开发环境——使用这些功能广泛的集成式开发环境，你可以非常容易地开始你的应用程序开发工作，这种开发环境能够满足你所有的要求，尤其是如果你拥有高速因特网访问条件时，更是如此。

如果你选择一个托管式开发环境，请使用你选择的编辑器，将一个简单的程序保存到你的Web服务器中。若要运行这个代码，你必须获得一个VSP账户，请向开发人员程序注册并且建立你的应用程序的URL。然后拨打VSP提供的电话号码，必要时，请使用用户ID和口令。

如果你选择一个模拟开发环境，请使用你选择编辑器来编写一个测试程序。下载并安装一个模拟器和有关的支持工具。遵照供应商的说明来启动应用程序。

如果你选择基于Web的开发环境，请连接一个开发人员程序，访问开发站点并开始运行代码。

第4章 VoiceXML用户界面

如今，如果你查看关于新兴因特网技术的会议日程和培训计划，你将会看到许多有关用户界面的工作室、培训和讨论组。如果你查看因特网上的就业广告，你会发现人们感兴趣的一些技能，例如用户分析和任务分析，以用户为中心的应用程序设计，以及应用程序的可用性测试等。为什么如今人们对用户界面产生了那么大的兴趣呢？为什么有那么多的人认识到用户界面对新兴因特网访问技术（如VoiceXML）的重要性呢？

本章将要通过下面几方面的工作来回答其中的一些问题：

- 复习用户界面的基本概念。
- 对人们熟悉的用户界面概念与用VoiceXML实现的语音用户界面（VUI）的相关组件进行比较。
- 讲述为什么说VUI的设计是成功地开发VoiceXML应用程序的一个重要组成部分。

完整地介绍与语音用户界面相关的人的因素，这不是本书要讲解的内容范围。本书中讲述VUI的各章要达到的主要目的是使读者能够基本上了解影响VUI设计的一些关键问题，并且举例说明怎样在你的VoiceXML应用程序中解决这些问题。

许多读者可能急于转入学习本书中专门介绍VoiceXML编程的章节。

这样做是很危险的！

出色的用户界面设计是你的VoiceXML应用程序取得成功的关键。VoiceXML提供了一个开发交互式语音因特网应用程序的出色工具，但是它不是解决一切问题的灵丹妙药。它只能提供你设计和建立的用户交互操作方式。

为了避免返工，或者更糟糕的是用户拒绝使用你的应用程序，你应该学习有关VUI的知识。学习VUI的知识并在VUI的设计上花些工夫，将有助于你用最有效的方式开发有用的、吸引人的应用程序。

4.1 用户界面的基本概念

自从原始人类认识到某些棍棒比其他东西具有更大的击打力量开始，就出现了用户界面（UI）。从此我们一直在高尔夫俱乐部、家用电器设备、汽车和软件等领域寻找更好的用户界面的设计。

如果你回顾一下PC上关于UI的某些概念，你将会认识到如下所示的用户界面特性：

- 在体力和脑力上与一种产品打交道的方法。
- 认识一种产品作用的途径。
- 使得产品便于使用或者便于学习的产品特性。
- 导致你喜欢或者不喜欢某个产品的外观。

4.1.1 用户界面的组成部分

除非你采用一种高级的生活方式，否则你每天使用的硬件和软件产品所拥有的用户界面都需要进行某种形式的物理交互操作。我们通常将这些用户界面视为理所当然的东西。

在开始讲述用户界面的作用和特征之前，本节首先要复习一下你经常使用的某些用户界面。表4-1列出了一些你可能接触过的产品及它们的用户界面组成要素。

表4-1 用户界面组成要素的举例

产品类型	用户界面组成要素
书面材料	目录、索引、标题 表格、列表、图形、字体 封面材料、纸张颜色、纹理
工具、运动器材和消遣设备	标签、紧固件、控件 产品的形状、建筑材料、重量、锋利的边缘或尖头边缘 把手的形状、盖子的纹理、颜色
电器设备和家用电子设备	标签、指示灯、液晶显示器 把手、旋钮、拨号盘、锁 部件之间的空间关系
运输和建筑设备	标签、液晶显示器、图形显示、图标 大门把手、方向盘、按钮、旋钮 钥匙、室内装潢材料、油漆
娱乐设备和游戏	监视器、液晶显示器、触摸屏 图标、图形、颜色 扬声器、麦克风、操纵杆、触垫

请注意，你不必考虑用计算机和计算机软件的概念来更新你对用户界面的理解以及它们每天对你产生的影响。

用户界面的组成部分可以帮助你理解怎样通过提供或者隐含提供下列特性来使用产品：

- **任务支持特性**——提供任务支持的用户界面组件能够回答下面这样一些问题，比如“该产品打算完成什么任务？”，“该产品怎样帮助我完成某项任务？”，“产品的形状或设计能够告诉我怎样使用该产品吗？”。
- **导航特性**——具备导航特性的用户界面能够回答下面这样一些问题，比如“我应该从什么地方开始操作？”，“下一步我应该进行什么操作？”，“我如何知道什么时候完成了我的工作？”，“如果我的操作出现错误，怎么办？”，“我应该如何退出操作？”。
- **满意度**——为用户提供满意度的用户界面组件将会下面这些问题的答案，比如“我喜欢使用本产品吗？”，“该产品看是去‘适合’我吗？”，“接触或者观察该产品使你感到愉快吗？”，“该产品容易使用吗？”。

4.1.2 CHUI、GUI和WUI

当我们听到“用户界面”这个术语时，许多人首先想到的是PC的图形用户界面（GUI），这说明有几代人都在阅读本书。我们很难记得PC的问世只有大约25年时间，而图形用户界面问世

的时间甚至更短——大约为15年。

在PC投入实际应用的这个比较短的时间里，它的用户界面已经经历了下面几个演变发展阶段：

1) 基于字符的用户界面——应用程序给出一个光标，用户键入一些多少有点儿像密码的命令，其输出可以显示在屏幕上，也可以打印成文本。

2) CHUI（基于字符的用户界面）——应用程序使用基于字符的符号来组成屏幕上的对象，使它们在视觉上更加美观。输入主要是键入的文本。输出可以显示在屏幕上，也可以打印成文本，有时在屏幕上增加一些简单的图形，以便改善输出的效果。

3) GUI（图形用户界面）——应用程序使用以图形表示的窗口、图标和菜单。可以用指向设备来选择需要的选项。图形用户界面也称为WIMP（即窗口、图标、菜单和指向设备）用户界面。它支持多种输入和输出方法。

尽管PC的图形用户界面问世的时间不长，但它已经强烈地影响到了我们对如何与技术打交道的期望值。

4.1.3 Web用户界面

近年来，WIMP图形用户界面的概念已经应用于基于浏览器的Web应用程序的用户界面，结果产生了Web应用程序用户界面（WUI）。WUI及其相关的多媒体组件导致人们希望获得访问因特网内容所需要的高带宽。本书中介绍语音用户界面（VUI）的各章其目的是展示如何将你已经了解的关于GUI和WUI的知识与VoiceXML这样的功能有限的技术关联起来。VoiceXML用户界面的功能是有限的，因为它不能像Web图形用户界面这样的高带宽多媒体通信客户程序那样传递大量的信息。

4.2 VUI概述

用户界面是用户与之进行交互操作的产品的一个组成部分。由于本书介绍的是如何开发VoiceXML应用程序，所以我们必须介绍语音用户界面（VUI）。VUI可以是VoiceXML应用程序的任何一个部分，用户必须与它进行交互操作，以便执行应用程序支持的各项操作任务。

VUI和VUI的设计与VoiceXML应用程序如何执行下列任务有关：

- 让用户了解他们的输入是必须的。
- 当用户对着电话送话器说话或者按他们的电话号码键盘时，接收和识别用户的输入信息。
- 通过用户的电话听筒向用户提供输出信息。

采用专用电话应答系统、基于PC的听写工具、自动电话接线员等形式的语音应用已经出现了多年了。开发这些系统的专家已经提出了许多术语来指称语音应用程序的用户界面，比如语音用户界面（SUI）和听觉用户界面（AUI）等，这些术语表达了下面这些基本概念：

- 人们向应用程序用户界面说话，或者使用电话号码键盘向它发送声音（输入）。
- 应用程序用户界面借助声音将信息反馈到用户耳中（输出）。

4.2.1 VUI的功能

VUI对于它的用户来说代表了一个VoiceXML应用程序。VUI起码应该具备下列功能：

- 用便于理解的方式向用户说明应用程序是怎样工作的，以及它支持什么样的操作任务。
- 采用口语表达的方式或者通过按电话号码键盘上的按钮产生的DTMF（双音多频）声音来接收用户的输入信息。
- 将系统输出传递给电话听筒。
- 支持用户完成操作任务。
- 支持用户错误或系统错误的恢复。

4.2.2 VUI的术语

我们必须对VUI和其他交互式语音应答应用程序有关的术语有所了解。表4-2列出了VUI的术语。

表4-2 VUI的术语

术 语	定 义
ASR（自动语音识别）	能够对口语发音进行分析，并且识别指定的单词（参见grammar条目）。语音识别可以受说话人的制约（当经过练习适应用户的声音后就能够进行语音识别），也可以是不受说话人的制约（不需要根据支持的语言的典型说话人的练习就能够进行语音识别）
声音沙漏图标	目前VoiceXML主要用于不受说话人制约的平台
声音图标	代表正在进行的系统活动的一个声音
打断	代表报警信息或商标信息的一个声音
DTMF（双音多频）	通过说话或按电话号码键盘上的按钮来中断应用程序输出的一种功能
Earcon	由电话号码键盘产生的一种音频
全局导航、全局命令	参见声音图标
语法	用户始终都可以使用的命令
本地导航、本地命令	ASR引擎识别为用户有效输入的一个单词或短语的列表。一个应用程序可以使用多个语法
定向的机器	只有当用户位于应用程序的某个特定位置时才能使用的命令
混合主动式	参见系统主动式
提示	一个VoiceXML对话框，它既允许应用程序也允许用户来确定下一步将会发生什么情况
录制的通告	你的应用程序“说的”一些短语，用于提示用户向应用程序提供输入信息
STT（语音文本转换）	在应用程序输出期间向用户播放的一种预先录制好的通告
系统主动式	将语音变成文本单词
TTS（文本语音转换）	一个VoiceXML对话框，在这个对话框里，应用程序主动启动交互操作，并且将应答的范围控制为对话框要求的那些项目
语音录音	将文本变成模拟语音
	录制用户的输入信息并将信息存储起来供将来输出的功能

4.2.3 VUI的输入/输出

下面让我们来看一看VUI输入和输出的基本情况。表4-3对VUI的输入/输出方法做了一个介绍。

表4-3 VUI的输入/输出举例

交互操作的方向	VUI组件举例	功 能
用户输入	代表一个单词或短语的口语发音	为应用程序提供输入。如果符合下面的条件，那么这个输入就能够成功地支持应用程序的按步骤运行：它包含在一个有效的语法中，并且被ASR所识别 举例：“What's the weather in Tucson?”（塔克森的气候如何？）
	代表一个数字的口语发音	当输入可以是一个口述的数字或DTMF击键时，便向应用程序提供输入 举例“One”（1）
	DTMF击键	当必须使用DTMF或者DTMF作为一个选项被系统支持时，便向系统提供输入 举例：“Hi. My name is Fred. I like romantic dinners at French restaurants.”（你好，我是Fred，我喜欢在法国餐厅享受浪漫晚餐。）
	其他指向VUI的语音	为一个应用程序提供输入，该输入被录制好后由同一个应用程序或另一个应用程序将来播放 举例：“<prompt>Say the title of the book you're looking for.</prompt>（请说出你要找的图书的名字）
应用程序输出	提示	向用户“说话”，提示他们向应用程序提供输入 举例：“<prompt>Say the title of the book you're looking for.</prompt>（请说出你要找的图书的名字）
	声音	应用程序可以传递来自提示内部的声音回放。声音输出的例子包括录制好的通告，音乐，语音沙漏标和声音图标 举例：“Do you want weather sports, or stock quotes?”（你是想要天气预报、体育消息还是股票行情？）
	TTS（文本语音转换）	应用程序可以使用TTS传递来自提示内部的文本信息 举例：“Do you want weather sports, or stock quotes?”（你是想要天气预报、体育消息还是股票行情？）
	菜单	菜单能够在提示内部将一个特定选项的列表传递给用户 举例：“Do you want weather sports, or stock quotes?”（你是想要天气预报、体育消息还是股票行情？）
窗体		VoiceXML窗体在VUI的后面运行，用于收集应用程序需要的数据 例如，“What is your age?”（你几岁啦？）是用户听到的一个提示。对应的数据字段由窗体属性来定义

下一节将要对VUI和Web GUI提供的功能作一个比较。

4.3 VUI与GUI在设计上的差异

对于从事VoiceXML应用程序开发的许多人来说，VUI是一个全新的概念。VUI与你过去已经设计过的其他类型用户界面之间的主要差别包括以下几个方面：

- VUI是不可视的用户界面。就用户而言，VUI只存在于他们的脑海之中。VUI必须设计成尽可能不需要用户进行记忆和复杂的识别过程。这是VUI与GUI之间最主要的区别，并且由此而带来了其他许多方面的设计差异。
- VUI属于单模界面。从实际应用上讲，对VoiceXML应用程序的惟一输入是声音，并且声音是惟一的输出介质。单模界面不能像WIMP GUI之类的多模界面那样有效地与用户进行通信。
- VUI必须做到任务集中明确。所有用户界面都应当做到任务集中明确，但是由于VUI的功能有限，因此它需要采用一种简单的工作流程，尽量减少分支流程。紧凑而任务明确的应用程序能够保持用户的兴趣，并且能够使用户了解他们的应用程序运行到了什么位置。
- VUI可以用在能够吸引用户的注意力和不需要复杂的识别过程的环境中。从某种程度上讲，VoiceXML应用程序是为了适应移动中的人们的需要而开发的。在驾车、走路和工作中使用的应用程序必须做到非常简单。它的提示必须设计成尽可能不要求用户进行记忆。必须考虑到并且避免应用程序中可能导致错误的各种条件。
- VUI中目前还没有与GUI相对应的某些标准元素，比如Home、Back、出错对话框和在线帮助等。你必须创建相应的VUI结构和相关的应用程序处理方式。

4.3.1 Web浏览器用户界面与VUI的比较

为了与VUI进行比较，让我们观察一个用于访问因特网内容的GUI例子，这是Netscape Navigeter浏览器窗口中显示的一个有代表性的Web站点。

图4-1显示了用Netscape Web浏览器看到的一个简单的Web网页。与浏览器显示有关的屏幕内容可以分成3个基本区域，如图所示：

- 1) 围绕着浏览器的操作系统窗口“框”。
- 2) 浏览器软件组件。
- 3) 页面显示区。

每一个基本区域都从许多方面影响到了用户界面，你必须对这些方面有所了解，以便将你对Web GUI应用程序了解的知识转换成新的因特网访问方式，例如VoiceXML。

1. 窗口对用户界面的影响

图4-1中标号为1的操作系统窗口将对用户界面产生下面几方面的影响：

- 它提供了一个用于定义应用程序边界的框。这样你就不会将桌面上的对象与Web应用程序相关的对象相混淆，因为有了边界后，你就可以知道在屏幕上的应用程序从何处开始，到何处结束。
- 它提供了一些控件，你可以用来改变显示对象的大小，也可以用来完全退出应用程序。

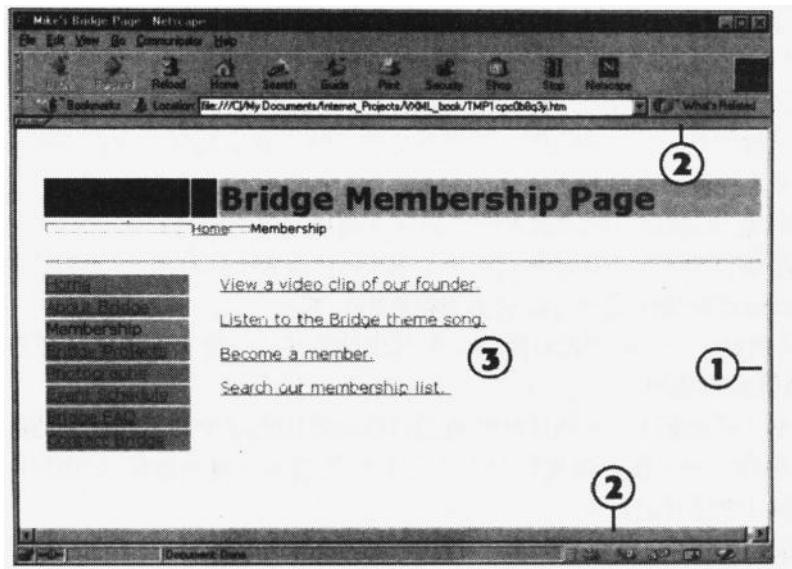


图4-1 用Netscape Navigeter浏览器显示的Web页

- 它提供了一个名字，使你能够清楚地了解你正在窗口中进行什么操作。
这个操作系统窗口可以作为你使用应用程序时的一个参照点和控制点。

VUI的窗口对用户界面的影响

VoiceXML应用程序至少必须提供由操作系统窗口提供的某些功能。例如，VUI应当使用户能够：

- 了解应用程序究竟能够做些什么。
- 方便地执行应用程序的退出功能。
- 了解他们正在使用自动运行的应用程序。

2. 浏览器对用户界面的影响

图4-1中标号为2的特定浏览器元素提供了下面这些用户界面组件：

- Home——将用户转至一个已知的参照点，这个参照点与用户当前在Web应用程序中的位置无关。
- Back——使用户返回上一个操作步骤。
- Bookmarks——为用户提供一种方法，使他们能够存储和访问重要的或常用Web网页。
- Location——向用户显示在Web站点或应用程序中他们的当前位置的URL。
- Status bar——向用户确认应用程序是否正在运行。

VUI的浏览器对用户界面的影响

显然，VUI不能向用户提供可视信息。此外，目前VUI还没有与Home或Back等价的标准组件。当你设计VoiceXML应用程序时，你必须在语法中加上与导航相关的术语，然后提供相应的软件，以便以对用户有意义的方式对这些术语作出“应答”。

至少你必须提供下列等价组件：

- Home的等价组件——使用户返回某个已知起始点（比如你的应用程序的起始点或语音门户的主菜单）的术语或短语。

- Back的等价组件——使用户每次能够后退一步的术语或短语。

- Statue的等价组件——能够使用户了解系统状态的声音沙漏标，声音图标或录制好的通告。

3. 页面显示区对用户界面的影响

图4-1中标号为3的特定页面元素也能够提供下列用户界面组件：

- Home（起始位置）——在这种情况下，Home代表Web站点或应用程序的起点，而不是浏览器的Home按钮所代表的与应用程序相关的位置。
- Content（内容）——用于表示网页的作用或目的并且支持页面操作任务的图形、声音、文本、视频信息和动画。
- Hyperlinks（超链接）——将用户的转到站点或应用程序中的不同位置的链接。
- Forms（窗体）——负责接收用户的输入并且将输入信息传输给应用程序以便进行处理的组件（图4-1中没有说明）。

VUI的页面显示对用户界面的影响

你可能已经注意到Home用户界面的功能可能出现的混乱状态。在Web图形用户界面的例子中，Home可以指与应用程序无关的一个位置（浏览器的Home按钮），页可以指应用程序中的一个起始点（页面显示区中的Home按钮）。你的VUI必须支持语法中的下面两个不同的home等价术语：

- 一个是像“Go Home”这样的术语，用于返回语音门户。
- 另一个是像“Start Over”这样的术语，用于返回你的应用程序的起始点。

VUI不能传递与Web图形用户界面相同的那些内容类型。因此你必须根据声音通信信道的有限功能，降低对你的VoiceXML应用程序所支持的功能的期望值。

VUI能够支持下列等价组件：

- Hyperlinks（超链接）——本组件允许用户在你的应用程序中随意移动。
- Forms（窗体）——本组件用于定义用户的输入数据并控制应用程序对输入数据的使用。

4.3.2 Web GUI与VUI的比较总结

表4-4对Web应用程序的GUI与VUI的组件及其特性进行了比较。

表4-4 Web GUI与VUI的对照表

Web GUI的组件或特性	VUI的对应组件或特性	VUI的举例或说明
浏览器导航工具栏上的Back按钮	一个语音命令，用于使用户返回上一个操作步骤	“Go Back”，配有支持的语法和编程方法
浏览器导航工具栏上的Home按钮	一个语音命令，用于将用户转到一个已知的起始点，比如一个语音门户服务程序的主菜单	“Go Home”，配有支持的语法和编程方法
浏览器窗口中的Home链接（Web页的一部分）	一个语音命令，用于将用户转到当前应用程序中的一个已知起始点	“Start Over”，配有支持的语法和编程方法

(续)

Web GUI的组件或特性	VUI的对应组件或特性	VUI的举例或说明
屏幕的布局，包括基本颜色、标记和图形的主题	录制好的通告和TTS语音，发音的风格、声音的粗细和声音的特征	举例： 能够说明VoiceXML页的作用的提示 选择一种目标用户群体所喜欢的声音。许多北美用户通常喜欢女低音 使用针对应用程序的目标用户群体的俚语或土语 每次进入应用程序时播放一组声音
用于指明错误和错误恢复的弹出功能或新的Web页	用于指明错误和错误恢复的声音、TTS或者录制好的通告	举例： “Sorry, I didn't understand that” (对不起,我不明白你的意思)。请用下面这样的格式说年月日，“August second, two thousand one”
Help链接或在线链接，上下文敏感的帮助页	Help信息	VUI帮助信息与出错信息可以互不干扰地放在一起,就像上面这个例子中的情况那样 举例： 使用混合主动式语法,可以使用户转到其他应用程序或某个应用程序中的某些模块
链接到其他Web页	对系统函数进行编程,以便对提示的输入作出应答	VoiceXML表单能够支持对字段和变量的定义,获取字段输入并且将逻辑方法与窗体元素获取的用户提供的数据关联起来
表单输入	与提示相关的VoiceXML表单	举例： 一些平淡无奇的声音,比如在水中行走的声音,冒水泡的声音等等
选项列表、单选按钮 Web应用程序内部产生的状态指示和其他用户反馈信息	参见“表单输入” “声音沙漏”的音调,声音、音乐或消息,用于说明系统仍在运行	

4.4 VUI用户的特点

VoiceXML应用程序应该面向明确定义的用户群体,但是VUI的用户有一些常见的特点,这有助于你了解他们与PC的GUI用户之间存在的区别,这些区别包括以下几方面:

- 他们对PC和因特网的使用经验有限。
- 他们处于移动环境之中。
- 他们采用单一的I/O模式。

4.4.1 对PC和因特网的使用经验有限

对于世界上大多数不拥有自己的PC机或无法访问因特网的人来说, VoiceXML应用程序最强的一个功能就是它们可以用来访问因特网。但是,要想得到这些新用户,你就必须满足新的用户界面的需求,而许多开发商本来并不需要满足这样的需求。

为没有因特网访问经验或PC使用经验的用户群体开发应用程序,这对你来说将意味着什么

呢？下面我们举几个例子：

- 用户对VoiceXML应用程序的最初认识可能不准确，因为他们是根据复杂的人与人之间的对话模式来思考问题的。
- 用户对VoiceXML应用程序的最初认识可能不准确，因为他们是根据以前使用规范的交互式语音应答（IVR）菜单的经验来思考问题的。
- 用户不了解计算机网络，所以，对于根据他们了解的访问远程计算机的速度和可靠性的期望值，他们不能进行相应的调整。
- 用户对应用程序的初始认识不是根据Home、Back、超级链接、和基于字段的窗体等与因特网相关的概念产生的。

尽管VoiceXML应用程序的许多早期用户将是这种技术的最早使用者，但是他们只是潜在的VoiceXML应用程序全体用户的冰山一角。了解你的应用程序的设计对一个不一定懂得使用PC的人的影响是非常重要的。

4.4.2 处于移动环境之中

人们使用移动电话来访问新技术提供的服务，如VoiceXML应用程序。你必须了解移动式访问对应用程序的设计产生的相关影响。

在进行应用程序的设计和开发过程中，你应该考虑的移动式访问的特点有以下几个方面：

- 人们在行走和驾车的时候使用移动电话。他们很少有时间来考虑如何使用的你应用程序。
- 移动式环境常常十分嘈杂，并且会因为跨区转接而使电话中断。从语音识别系统的目前状况来看，移动电话用户访问语音识别系统时，复杂的应用程序难以有效地运行。
- 访问移动式服务的人常常显得急急忙忙。他们往往希望使用特定任务的应用程序，以便迅速完成他们的工作。
- 使用移动电话访问VoiceXML应用程序的人通常希望能够用一只手来打电话。因此他们很难并且不可能用电话键盘来输入数据。

4.4.3 单一的输入/输出模式

VoiceXML应用程序只能使用声音介质来提供输出和接收输入。仅仅使用声音来运行的应用程序所传递的信息不能像使用多种介质的应用程序（如Web GUI应用程序）那样多。

当声音作为传递信息的惟一介质时，对应应用程序设计的影响将与这样一个情况有关，即用户不可能记住很多的信息。因此VoiceXML应用程序必须做到任务集中、简单，并且不要求用户为了使用这种应用程序而进行任何学习。

4.5 设计VUI时应该考虑的其他重要问题

前面几节内容列举了VUI必须支持的重要功能以及设计时应该考虑的一些重要问题。此外，出色的VUI设计还应该达到以下几方面的要求：

- 使VoiceXML应用程序能够吸引目标用户群。
- 能够代表目标用户群希望执行的任务。

- 提供一个能够引导人们快速而有效地进行代码开发的结构。
- 尽量减少平台和系统给用户使用VUI带来的影响。

4.5.1 吸引你的目标用户

成功的VoiceXML应用程序总是将注意力牢牢集中在目标用户希望进行的活动（任务）上。这就意味着在你为VoiceXML应用程序选择它要执行的操作任务之前，你必须知道你的用户是些什么人，他们想要用它来做什么。

当目标用户确定之后，就要靠VUI的良好设计来以有效而吸引人的方式向用户传递信息了。为了使你的VoiceXML应用程序能够更好地吸引目标用户，你可以调整的VUI组件包括以下几个：

- 声音。它可以用于录制好的通告和进行TTS播放，声音可以因为性别、声调和速度的不同而不同。应该选择能够在目标用户群中产生信任和让人喜爱的声音。
- 对话框的样式。提示，录制的通告和其他系统输出应当设计成符合目标用户群的文化特点。
- 应用程序的速度。系统计时器和应用程序的暂停可以进行调整，使应用程序的运行速度最适合任务的执行、典型的应用环境和目标用户群体的需要。

4.5.2 能够代表目标用户群希望执行的任务

在你为VoiceXML应用程序选择它要执行的任务之前，你必须了解你的用户是些什么人，他们想用它来做什么。选定任务以后，就要靠VUI用一种便于用户理解的方式来向用户展示它能够执行的任务。为了使VUI能够很好地向用户展示它能执行的任务，可以采用下面几种方法：

- 将VUI的设计建立一个经过良好分析的工作流基础上。
- 根据最实用的方法来设计有效的提示。
- 使用用户熟悉的术语。

4.5.3 提供一个适合代码开发的结构

作为以用户为中心的应用程序设计过程的一部分而开发的出色的VUI设计，能够提供一个很好的代码开发结构，它能够缩短代码开发的时间，并且减少应用程序开发过程中的返工现象。

除了上面介绍的VUI能够执行的任务外，VUI的设计还通过提供下列条件来支持快速代码开发：

- 应用程序的初始原型。
- 可以直接放入应用程序代码的提示内容。
- 通过应用程序的各种路径的一个清晰视图。包括与任务的执行相关的转移，全局导航和本地导航，访问帮助信息，以及错误的识别与错误的恢复。

我参加过的一些开发项目说明，当VUI作为以用户为中心的总体设计和开发工作的一部分来进行的时候，上面列出的这些条件能够使代码的开发周期缩短30%到50%。

4.5.4 尽量减少限制条件造成的影响

一个好的VUI设计能够最大限度地减少技术、系统和平台等限制条件带来的影响。这些限制因素与下列因素有关：

- 语音识别——处于当前演变发展状态下的语音系统所固有的一些问题。
- 平台的功能——功能不断变化，VoiceXML应该适应不同的语音系统平台的要求。
- 网络——与VoiceXML应用程序采用的结构相关的等待时间和外部系统故障。

VUI的设计能够减少各种限制因素给用户对应用程序的认识带来的影响。例如，VUI的设计可以：

- 采用提示的设计和DTMF输入来克服ASR的缺点。
- 采用“声音沙漏标”的声音或有趣的提示来减少用户因为平台或应用程序的等待时间过长而放弃使用VoiceXML应用程序。
- 采用系统计时器和给用户发送消息，减少用户因为外部系统的等待时间过长或者发生故障而对应用程序产生的不良感觉。

4.6 小结

从地球上出现人类起，用户界面就问世了。任何时候只要人与产品或服务打交道，他就要与产品的用户界面进行交互操作。

由于用户界面代表了提供给用户的产品或服务，所以用户界面的设计是应用程序开发的一个重要组成部分。

用户通过语音用户界面（VUI）与VoiceXML应用程序进行交互操作。VUI传递的信息不像GUI那么多，但是有些VUI组件与Web GUI组件是类似的。这些组件包括Home和Back按钮、超级链接及数据收集窗体等。

VUI的设计对于VoiceXML应用程序的成功与否是至关重要的，当VoiceXML应用程序作为以用户为中心的设计过程的一部分完成时，就可以最大限度地减少系统限制因素产生的影响，也有助于代码的有效开发，并且可以缩短产品上市的时间。

接下来的几章将要详细介绍本章抛砖引玉讲述的关于VUI的一些问题。

第5章 VUI设计的概述

第4章对用户界面进行了概述，并且介绍了使用VoiceXML来实现的语音用户界面（VUI）。它还介绍了与VUI的设计有关的许多问题和限制因素，并且介绍了与VUI的设计有关的VoiceXML元素的例子。

本章的重点将从确定问题转到说明有关的解决方案。

遵循一定的工作过程是使工作能够反复取得成功的重要先决条件。这种过程将使你在转向新的工作领域时能够迅速学习到有关的知识。由于VUI的设计对大多数人来说还是一项全新的工作，因此本章将重点从开发过程的角度来介绍VUI的设计。VUI设计的过程将为你提供迅速开始VUI设计工作所需要的知识和实现基础。

2000年期间，我必须着手加快进行用于两种因特网访问技术的用户界面的设计，这两种技术是VoiceXML和WML（无线标记语言），这是一种移动电话因特网浏览器使用的语言。我能够比较迅速地转入这些新技术开发的原因之一是我们原本就具备进行以用户为中心的界面设计（UCD）过程的经验。这个经验也适用于更新技术的开发。本章将要介绍如何将UCD过程用于VUI的设计。当你按照这个设计过程来工作时，你将要了解VUI的用法和怎样将它应用到VoiceXML应用程序中去。

5.1 软件开发过程的概述

由于你正在阅读本书，因此你可能对应用软件的开发有一定的了解。图5-1显示了软件开发的通用进程。

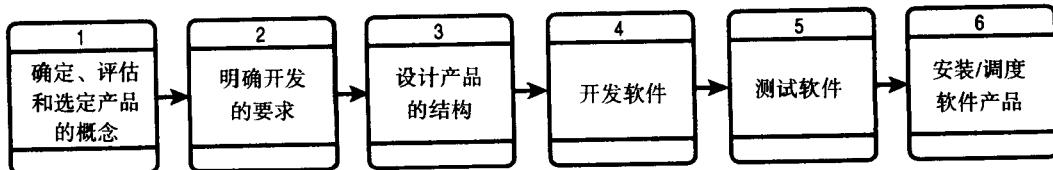


图5-1 高级软件产品开发过程的流程图

高级软件产品的开发过程所包括的各个步骤和阶段将负责进行下列工作：

- 确定和选定产品开发的时机。
- 确定产品和结构的要求。
- 设计产品和结构。
- 开发产品。
- 测试产品和相关的平台。
- 安装或部署产品。

许多公司和个人都发现，虽然他们遵照了一定的设计流程进行设计，但是结果设计出来的

应用程序仍然不受欢迎，难以使用或者学习，甚至在向用户每推出一个版本后都需要进行重大的返工。造成这种情况的原因往往是因为在开发过程的各个阶段没有对用户的需求和环境给予足够的重视。

5.2 UCD过程的概述

如果说有些人执行软件开发过程的方法存在缺点的话，那么必定存在解决这个问题的办法。不是吗？近年来，UCD已经成为与应用程序开发相关的许多问题的最佳解决办法。UCD过程被许多软件开发公司视为业界的最佳方案。

从用户、用户的任务和用户界面的角度出发，通过强调产品的设计和开发，UCD的侧重点在于优化用户的使用经验上。

UCD过程从应用程序开发的前期工作开始，努力确保开发出合格的产品。在软件开发的整个过程中，他们始终不断地展开产品的可用性评估和测试，以提高产品的可用性，不让问题存在于代码之中。而且在开发过程结束时，他们仍然要进行产品的可用性测试和客户满意度调查，用以对产品进行最后的评估，并且为产品的下一个版本收集意见。

在过去的几年里，UCD过程已经成为众人注目的焦点，因为它们支持人们所喜爱的高质量应用程序的设计和开发，符合人们对缩短开发周期和降低开发成本的要求。

最近的情况表明，人们强烈希望将UCD过程的某些方面纳入概念的定义和概念的选择过程，以便选择出用户喜欢的、“适销对路”的产品进行研发。

UCD过程改进了软件开发过程产生的结果，因为它能够确保目标用户的需求影响下面的工作：

- 定义和选择新产品的概念。
- 产品的要求。
- 产品的重新设计的工作流程实现自动化。
- 用户界面的设计。
- 应用程序的评估。
- 纳入未来产品版本的特性。

UCD过程将以前分散的工作方法整合在一起，它们包括以下几个方面：

- 软件产品的设计、开发和测试。
- 用户界面的设计、开发和测试。
- 操作技术（人对作业的操作）。
- 人的因素。
- 技术文件的编写、培训方案的制订和培训的提供。
- EPPS——电子性能支持系统的开发。

通过将以前用分散方法从事的工作加以合并和压缩（或根本不进行合并和压缩），UCD过程使你能够更快开发出更有可能产生效益的应用程序，因为这些应用程序容易使用并且能够满足用户的需求和愿望。

UCD过程提供的一套方法有助于你解决与VUI开发相关的问题和新课题。

图5-2描述了一个通用的UCD进程，并且说明了它与图5-1描述的软件开发过程之间的关系。

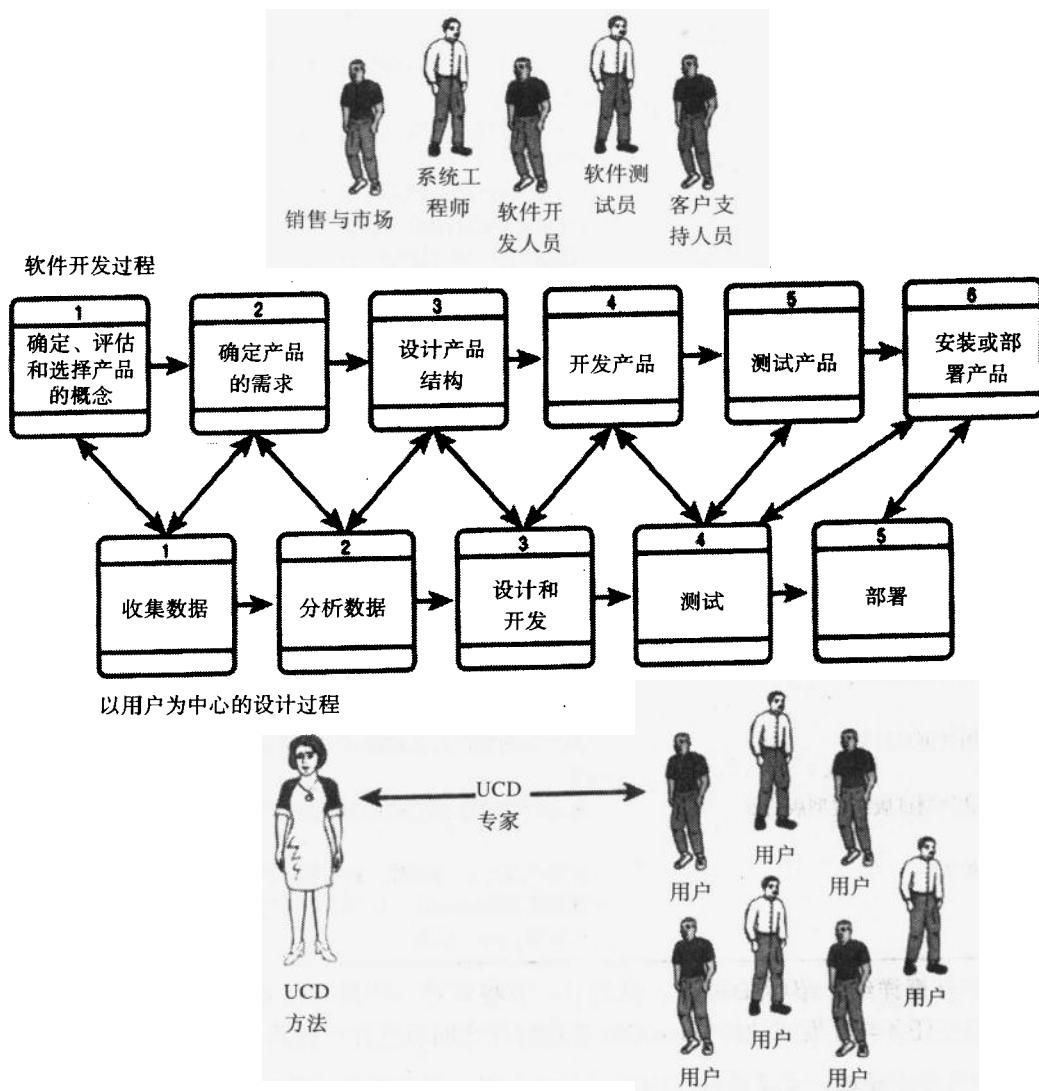


图5-2 UCD过程提高了用户进行软件开发的能力

表5-1列出了通用UCD过程中包含的典型任务。

表5-1 UCD任务的概述

任 务	说 明
用户分析	采用调查、采访或其他研究方法来搜集数据，以便确定影响产品设计的用户特点
任务分析	对工作过程的数据进行分析，确定目前的工作流程，并且

(续)

任 务	说 明
上下文查询	确定通过任务的自动化或帮助任务的执行来更改用户工作的可能性
分析数据	访问用户的工作站点，以便通过观察来搜集用户、任务和环境数据
确定任务环境和/或过程图	使用相似性分析方法、相互关系的有向图和其他分析工具来评估用户数据和任务数据
确定可用性要求	绘制用于说明用户任务过程的高层流程图，以便与开发人员和系统工程师展开讨论
设计和开发低保真度的用户界面原型	确定可用性的要求和目标，以便将其纳入系统要求之中 低保真度原型通常是一种不能实际运行的产品原型，用于说明该产品应该如何运行，比如用书面文字编写的语音用户界面或原型工具，就属于这种原型
选择HPS支持特性	为用户界面选择HPS（支持人工操作）方法和元素
设计高保真度的用户界面原型	设计高保真度的产品原型，并且与开发人员合作开发该产品的模型
重复进行原型的可用性测试	对原型上进行可用性测试（使用伪用户和/或真用户），提供可用性的测试原则性评估
用户界面实现的协作	观察正在进行的开发工作，并且与开发人员讨论用户界面的实现方法
重复进行可用性测试	根据需要重复进行可用性测试，与开发人员合作，设计解决可用性问题的方案
制订可用性测试计划	为产品的客户版本制订可用性测试方案和进行用户满意度调查
进行产品的测试版或定型版的可用性测试	进行产品的可用性测试和满意度调查
编写UCD报告	汇总产品的UCD过程，获取设计数据和设计方案、可用性测试结果和解决办法、用户满意度结论，并且将这些数据纳入产品的下一个版本

本书并不打算详细介绍UCD过程，但是下一节将要对一些具有代表性的任务作一个概述，并且要说明这些任务与开发成功的VoiceXML应用程序之间有些什么关系。

注意 如果你希望进一步了解关于UCD进程的知识，那么有许多很好的资源可以供你利用，比如本章结尾处的“UCD和系统设计参考资料”一节列出的若干资源。

以用户为中心的设计任务

下面各节将要详细介绍图5-2中显示的UCD过程的各个阶段，并且要讲述在每个阶段中你需要执行的任务。

第1阶段：数据搜集

第1阶段的任务是搜集信息，以确保你开发的应用程序是用户需要和想要的。

在第1阶段中，担当UCD专家职务的人负责搜集有关用户、用户任务、用户工作过程和用户

工作环境等方面的信息。数据搜集的方法包括上下文的查询（现场观察）、采访、对重点人群的考察、调查以及用户分析和任务分析。

在第1阶段中进行的工作要达到以下几个目的：

- 确定你重点针对的用户的特点，以便使你的应用程序具有吸引力并且成为用户想要的产品。用户的特点包括年龄、性别、文化水平和受教育程度。
- 确定用户想用你的应用程序做什么，或者是作为其他应用程序的概念来使用。这些事情的例子包括查看股票价格或旅行计划、采购任何东西和检索信息等。
- 确定用户目前执行的任务，这些任务与你想要使用新的应用程序来建立、激活或实现自动化的任务最相似。例如，他们现在可能通过登录到Web站点，阅读报纸，或者打电话给它们的经纪人来查看股票的价格。
- 确定用户应用产品时的环境特点，这些特点将会影响产品的设计、平台或结构。应用环境的特点的例子包括，噪声的量值、照明的亮度水平和存在竞争性任务。

第1阶段的成果包括，用户观察记录（如果进行了上下文查询的话）、用户分析报告、任务分析报告和来自于重点人群的考报告或调查报告。

第2阶段：分析数据

第2阶段的任务是分析在第一阶段中搜集到的用户数据和任务数据，并且用它来指导应用程序的设计。在第2阶段期间，担当UCD专家职务的人员将负责进行数据的分析，以便确定或开发：

- 任务环境或过程的流程，用于描述应用程序将要激活的新工作流程。
- 根据建议的工作流程确定应用程序结构的作用。
- 应用程序的可用性和用户使用应用程序应该达到什么样的感觉。
- 能够展示系统功能和设计的低保真度用户界面的原型。

第2阶段的成果包括用于确定潜在应用程序模块的任务或过程的流程图、使开发商和潜在客户能够迅速了解用户使用应用程序的感觉的低保真度用户界面的原型、应用程序的可用性要求和目标等，这些东西可以作为用户界面的设计文档，也可以作为对应用程序要求的文档的组成部分。

第3阶段：设计和开发

第3阶段的任务是完成用户界面的设计，并且与软件开发人员合作，用代码来实现应用程序的设计。在第3阶段期间，担当UCD专家职务的人的任务包括：

- 为用户界面选择支持的用户操作的结构和元素。
- 设计一个信息的结构（如果需要的话）。
- 通常与软件开发人员协作开发高保真度的应用程序原型。
- 对可以得到的应用程序原型进行初步的可用性试验评估和可用性测试。
- 根据要求按软件开发进程中确定的应用程序限制因素对用户界面进行重新设计。
- 与软件开发人员一道，确保开发过程中不会忽略用户使用应用程序时应该达到的感觉。

第3阶段的成果包括高保真度的用户界面原型、符合标准的设计和信息结构文档，以及根据早期可用性测试和评估提出的应用程序指导性建议。

第4阶段：测试

第4阶段的任务是根据要求和设计文档对用户使用应用程序的感觉进行测试，并且与软件开发人员一道，根据软件和系统测试期间发现的问题，对代码进行修改加工。在第4阶段中，担当UCD专家职务的人员的任务是：

- 为最终产品设计可用性测试计划。
- 进行可用性评估和测试。
- 根据需要，按照软件和系统测试的结果，修改用户界面的设计。
- 根据软件或系统测试发现的问题，与软件开发人员一道，确保用户使用应用程序的感觉不会受到不必要的损害。

第4阶段的成果包括改进软件和系统测试计划、可用性测试计划和可用性测试报告。

第5阶段：部署

第5阶段的任务是对应用程序的可用性进行现场测试，搜集用户的满意程度数据。在第5阶段中担当UCD专家职务的人员的任务是：

- 设计用户满意度调查方案。
- 对现场部署的应用程序实施可用性测试计划。
- 通过可用性测试和满意度调查搜集用户的反馈信息。

5.3 可用性测试

这是本章的最后一节，它将要介绍与使用UCD进行VoiceXML应用程序设计相关的非常广泛的问题。本节将要介绍UCD的一个组成部分，即可用性测试，并且要讲述它对VoiceXML应用程序的设计的影响。

测试是一个指导原则。VoiceXML可用性测试是描述支持实现VoiceXML应用程序可用性方法的指导原则。

许多拥有可用性测试或评估经验的人都使用面向图形用户界面的可用性测试方法来进行图形用户界面的评估。本节将要介绍一组VoiceXML的可用性测试方法和“逆向工程设计”方法，以便说明它们对应用程序的设计会产生什么样的影响。换句话说，我们要预先介绍可用性测试的方法，并且设计你的应用程序，使之尽可能符合这些指导原则。这样，你就可以在实现可用性的过程中排除许多障碍，这些障碍将会导致你的应用程序设计进行返工。

5.3.1 VoiceXML的可用性测试

许多软件的作者、咨询人员和实践者已经为各种类型的用户界面开发了可用性测试的原则。其中最著名的一位就是Jakob Nielsen，他在《Usability Engineering》(可用性工程设计)一书中描述了一组10个PC图形用户界面的可用性测试原则。在我开始从事VoiceXML应用程序开发工作时候，还没有得到人们广泛认可的可用性测试的指导原则，所以，我建立了表5-2列出的一组可用性测试原则，用于VoiceXML用户界面的设计和评估。

表5-2 VoiceXML可用性测试原则举例

测试原则	说 明
尽量降低对用户的记忆和识别过程的要求	VoiceXML应用程序的任务结构和提示设计必须做到直观和简单，以避免对用户提出太高的要求
保持简单而自然的对话框	与图形用户界面相比，VoiceXML用户界面中的“对话框”这一术语更符合它的原义。用户可以有效地与应用程序进行对话。对话框应当便于操作，并且很容易被目标用户理解
提供清楚的导航和快捷操作方式	VoiceXML应用程序通常应该容易学习，新用户或不常使用该应用程序的用户往往需要简单的导航工具。这种导航工具对于有经验的用户来说显得不太方便。与使用标准导航工具相比，快捷操作方式使得有经验的用户能够更快地在应用程序中移动
提供一致的反馈信息和便于使用的错误处理工具	用户如果不能确定VoiceXML应用程序是否在运行，或者他们遇到故障被困在里面，那么他们会很快放弃这种应用程序。在某种窗体中，用户需要快速获得一致的反馈信息。应用程序应该尽可能避免出错，并且在错误无法避免的时候，实现方便的错误恢复
保持导航和术语的一致性	使用统一的命令、术语和任务结构。尽可能遵循DTMF命令的标准，与老式语音应用程序保持一致
对选择的声音和声音的个性进行评估	VoiceXML应用程序选择的声音和声音的个性应当符合目标用户群中大多数人的喜好。在北美地区，通常应该使用一种女低音的声音，声音要悦耳，但是又带有深沉的说话风格

本章的其余部分将要介绍表5-2中列出的测试原则如何应用于VoiceXML应用程序的设计，特别是当它与VUI有关的时候。

本章将要更加详细地介绍每个测试原则，并且要讲述如何将每个测试原则应用于两种应用程序的VUI的设计，这两种应用程序是：

- 约束型应用程序——受用户环境和/或平台局限性约束的应用程序。
- 无约束型应用程序——在使用的设计方法方面拥有更大自由度的应用程序，因为它们受噪声环境和/或平台局限性的约束程度比较小。

注意 与图形用户界面或Web用户界面的功能相比，所有VoiceXML应用程序的图形用户界面都要受它们自身的性质所约束。

5.3.2 测试原则1：尽量降低对用户的记忆和识别过程的要求

测试原则1是说你必须尽可能不要求用户进行记忆和识别。与计算机的情况一样，你的应用程序的用户在记忆力和处理能力上都是有限的。当人们使用VoiceXML应用程序时，他们很少将他们全部注意力和思考能力集中在应用程序的使用上。除了驾车、走路和工作的时候之外，其他时间里他们也要使用你的应用程序。

由于下面几个方面的原因，VoiceXML应用程序的运行环境可能是非常“嘈杂的”：

- 手机和通信系统产生的噪声。

- 为了吸引用户的注意力。
- 复杂的任务工作流程。

这些噪声源对用户的记忆和识别能力提出了更高的要求。

如果你忽略了测试原则1，那么在通常的使用条件下，你的应用程序将很难让人能够使用，所以他们就不会使用你的应用程序。如果你遵循测试原则1，那么用户将会认为你的应用程序是“容易使用”的。

将测试原则1用于VUI的设计

无论你打算开发约束型应用程序还是非约束型应用程序，实现测试原则1的核心方法都是一样的：

- 进行用户分析——搞清何时、何处、如何以及在何种情况下人们想要使用你的应用程序。
- 进行任务分析——搞清你要用你的应用程序实现自动化或者运行的任务，目前人们是如何执行这些任务的。
- 分析数据——将任务分成较小的、模块式单元。根据你做的用户分析，确定有多少任务你可以将它们合理地纳入应用程序里。你可能发现你的思路并不是建立VoiceXML应用程序的好方法。

1. 测试原则1对约束型应用程序的影响

VoiceXML应用程序是否能够取得成功，这在很大程度上取决于托管应用程序的平台所具备的语音识别能力。当VoiceXML应用程序必须在噪声较大的环境种运行时，语音识别能力就可能会下降。当平台的语音识别能力下降时，用户的记忆和识别负担就会变得过重，原因在于应用程序：

- 无法识别有效的输入，从而形成了出错的条件。
- 错误地识别了无效输入，从而将用户导航至他所不需要的内容。

另外，你运行VoiceXML应用程序的环境可能非常“嘈杂”，这种环境会使用户不能专心完成与VoiceXML应用程序进行交互操作的任务。

为了将你的应用程序限制在它的运行环境之中，你可以采用下面几种办法：

- 使用双音多频（DTMF）语法（口语或按键）来接收输入信息。虽然这种方法不像出色的语音识别应用程序那样在技术上十分可取，但是它能够在噪声环境中运行。
- 简化任务流程。取消大部分任务的转移，严格限制使用的选项。只支持符合你的用户要求的主要任务。
- 将全球定位系统（GPS）机制纳入你的应用程序，确保用户不是在开车的时候连接你的应用程序。这只是一个玩笑，但我希望你能够领会我的意思，开车时他们很容易分心。

2. 测试原则1对非约束型应用程序的影响

如果你有幸拥有一个出色的托管平台和一个比较安静环境，那么与测试原则1有关的问题就不同了，不过这些问题并不会消失。

在这样的情况下，记忆和识别的负担很可能是你的VUI设计造成的。除了对用户和任务进行分析外，你还需要比你的开发约束型应用程序的同事们更加注意应用程序的提示的设计和错误的处理。

为了尽可能降低你的应用程序对用户的要求，你可以采取下面几种方法：

- 精简任务流程。减少任务的转移，只支持主要的任务选项。支持用户要求的一组完整的任务，不要夹杂那些不必要的任务。
- 将语法限制为只包含支持任务所需要的那些术语。通过正确的提示设计来引导用户提供正确的输入（也可参见测试原则2）。
- 使用有效的提示设计来帮助对用户的导航（也可以参见测试原则2和5）。
- 对出错情况采用分层次的应答方式（也请参见测试原则4）。

5.3.3 测试原则2：保持简单而自然的对话框

目前的VoiceXML应用程序实现的对话框采用了几种基本的样式。有些对话框样式是限制性很强的约束型应用程序，它们很像早期基于DTMF的电话应答系统，而有些对话框样式则是相对复杂的非限制型的应用程序，它们必须接收和处理范围广泛的用户输入信息，例如“我应该如何帮助你”这样的应用程序。

不管应用程序的类型或托管平台的语音识别能力如何，应用程序对话框应当是：

- 尽可能简单，但使用上又要做到非常清楚。
- 便于用户理解，包括帮助他们了解什么时候轮到他们说话。
- 让用户始终都知道他们处在什么位置以及他们在应用程序中做什么。

将测试原则2用于VUI的设计

对话框的设计包含两个基本组成部分：

- 对话框的总体结构，用于使用户了解他们在应用程序中做什么。
- 对话框中的各个提示和应答的设计，用于建立应用程序的各种对话的内容。

你的应用程序的功能是否符合测试原则2，这在很大程度上取决于你是否深入了解你的用户将要用你的应用程序来执行什么任务。如果你不想了解这些情况，那么当你继续将本书读下去的时候，你遇到的问题就会越来越严重。对于像VoiceXML这样的窄带宽用户界面来说，了解你的用户和他们的任务是绝对重要的。

鉴于你知道如何使用你的应用程序来执行用户的任务，因此对于约束型或非约束型应用程序来说，对话框的设计是不同的。

1. 测试原则2对约束型应用程序的影响

介绍测试原则1的这一节曾经建议在噪声环境中使用DTMF（口语或按键）语法。不管你选择使用DTMF语法还是有限的口语语法，提示的设计必须与期望的输入相匹配。

为约束型应用程序设计提示时，你可以使用的方法有下面几种：

- 按照传统的交互式语音服务程序使用的提示标准进行提示的设计（参见“UCD和系统设计参考资料”一节中列出的标准）。
- 统一按照“Goal→ Action”这个提示结构进行设计（比如“To get a stock quote, press or say ‘one’ now”（若要了解股票行情，现在请按‘1’键，或者说‘1’））。
- 建立错误应答消息，以准确地指明什么是有效的输入。

2. 测试原则2对非约束型应用程序的影响

进行对话框的设计时，非约束型应用程序有利也有弊。即使是不太受环境和平台约束的应用程序也无法创造奇迹。实际上，有些时候你使用的可能是规模有限的语法和不完善的语音功能，所以对话框和提示的设计仍然很重要。

为相对不受约束的应用程序设计提示时，你可以使用的方法有下面几种：

- 在提示中使用统一的术语，并且使它们与你的语法中的术语保持严格的一致。不要为了“富于变化”而使用替代词汇。请使用“To save your recording, say ‘save’”（若要保留你的录音，请说‘保留’），不要使用“To save your recording say ‘submit’”（若要保留你的录音，请说‘提交’）。
- 使用的提示设计应该显示它能够接收什么样的输入或者限制输入的逻辑范围（比如“Do you want to save your recording?”（你想保存你的录音吗？））。
- 让用户从术语或类别列表中选择他们的输入，而不是只提示他们进行随心所欲的输入。
- 建立对错误的应答，提供可能是最正确的应答作为首选应答，也可以建议正确的应答类型或应答的格式。

5.3.4 测试原则3：提供清楚的导航和快捷操作方式

由于两个主要的原因（并非低劣的设计），造成VoiceXML应用程序的用户很容易迷失他们在应用程序中的位置（请回顾一下测试原则1）：

- 他们必须依赖自己的记忆力，而这种记忆是短暂的。
- 他们可能并不经常使用某个应用程序，所以他们没有真正掌握它。

你可以设计一些导航系统，使用户能够更加容易记住他们的位置，并且通过你的应用程序找到他们解决问题的方法。

将测试原则3用于VUI的设计

假设你可以使用UCD方法来解决许多VUI的设计问题，但是当你设计应用程序的导航系统时，你仍然需要解决记忆问题和不经常使用应用程序的问题。

另一个问题也与应用程序的使用频率有关。如果某个人确实经常使用某个应用程序，那将会怎么样呢？已经成为专家的用户会发现，为新用户和不常使用应用程序的用户设计的导航系统，不一定会降低他们通过应用程序的导航速度。

若要满足两种用户的需求，你的VUI设计应当包含以下的功能：

- 全球导航——在应用程序中的任何位置几乎总是可以使用的命令。
- 本地导航——只能在应用程序的特定模块中使用的命令，包括配有特定上下文帮助信息的错误恢复例程。
- 快捷方式——不使用应用程序通常的导航结构的方法。快捷方式包括没有发布过的全球或本地导航命令、对打断功能的支持、以及对语音命令的DTMF替代操作方式。

1. 测试原则3对约束型应用程序的影响

约束型应用程序要求其导航系统的设计应该类似传统的交互式语音服务的导航系统。由于约束型系统必须简单，因此它的导航系统只需要考虑满足简单的错误恢复和帮助系统的需要。

若要为约束型应用程序设计导航系统，你可以采用下面的几个方法：

- 沿用早期电话应答系统使用的全球命令标准（参见“UCD和系统设计参考资料”这一节列出的标准）。典型的全球命令包括返回上一步、返回已知起始点、获取帮助信息以及对真人说话等所使用的方法（如果有是话）。
- 沿用早期电话应答系统使用的本地命令标准。典型的本地命令包括用DTMF（口语或按键）的1代表“是”，用2代表“否”。
- 使用应用程序问候语来表示导航系统的一些重要功能。你可以在问候语中说出访问帮助信息、重来或者联系某个人时使用的命令。

2. 测试原则3对非约束型应用程序的影响

非约束型应用程序需要复杂的导航功能。这种类型的应用程序通常使用更人性化的对话模式。导航系统的设计一方面必须建立能够有效地使用可用技术的应用程序，另一方面又必须有效地支持使用应用程序的用户，两者之间不可偏废。

用于非约束型应用程序的导航系统常常采用“我应该怎样帮助你”的模式。这种导航模式提示用户输入有关的信息，然后由此而开始运行。由于我们一直使用不太完善的语音识别和有限的语法来进行工作，因此导航系统有责任（与其他VUI元素相结合）：

- 帮助用户了解他们能用这个应用程序做什么。
- 指导用户提供有用的输入。
- 对用户输入信息中含义不清楚的问题作出反映。
- 如果用户对操作完全搞不清楚的话，那么使用户返回到一个已知的起点。

为这种模式的导航系统进行设计时，你可采用下面的几种方法：

- 格外注意为提供全球导航和本地导航而使用的语法。语法的设计应该做到便于将它们与无导航的输入相区分。
- 在不能确定用户的输入表示什么意思的时候，根据最佳的猜测来进行错误恢复。例如，如果输入的信息是：“football”、“foosball”或“free for all”，让用户从这3个输入信息中选一个，那么正确性最大的一个输入应该是它们当中的第一个输入。
- 在错误消息中插入与上下文相关的帮助信息。例如，“I didn't understand the sport you named. Sports available for selection are. . .”（我不明白你说的运动项目是什么。可供选择的运动项目有. . .）。
- 在导航模式中建立返回已知点的功能，如果用户看起来搞不清楚的话，便使他们返回这些已知点。例如，经过3个错误恢复例程后，使用户返回当前模块的起始提示。

5.3.5 测试原则4：提供一致的反馈信息和便于使用的错误处理工具

这个测试原则与测试原则3紧密相关。虽然对用户愿意接受什么样的访问因特网的语音服务问题进行的正式研究并不多，但是现有的数据似乎可以说明用户在使用VoiceXML应用程序之类的服务时常常显得没有耐心。

提供用户反馈信息时通常可以接受的指导原则是：

- 在用户输入信息后作出反馈——通过转到下一个模块或传递错误恢复消息，在1-3秒时间后

提供反馈消息。

- 对超时作出反馈——在3-8秒后提供反馈。

请对你的用户测试你的反馈时间，不过上面的两个指导原则是很好的起点。

将测试原则4用于VDI的设计

没有耐心的用户通常的做法是挂断电话。你并不希望他们这样做，所以你需要使用测试原则4来确保你尽了最大的努力来避免使用户失望。你可以采用的方法包括：

- 在应用程序等待接收因特网内容时向用户提供反馈信息。
- 当外部系统产生时滞的时候向用户提供解释性反馈信息。
- 提供尽可能对用户有帮助和无碍的错误恢复例程。

1. 测试原则4对约束型应用程序的影响

即使是约束型应用程序，也必须向用户提供反馈信息和使用方便的出错消息。约束型应用程序可以根据应用程序和托管应用程序的系统的功能，用许多种方法来向用户提供反馈信息。本节中将要介绍提供反馈信息的最简单的方法。其他方法将在非约束型应用程序中介绍。

你可以用来提供反馈信息的方法有以下几种：

- 对有效输入的反馈通常是直接进入到执行应用程序中的指定功能。
- 对有效输入的反馈信息可以加以扩充，以便包含已经进入的模块的名字。例如，如果用户提供的有效输入是“football”，那么当应用程序进入该模块时，它就会说出这个“football”词汇。
- 对因特网访问的等待时间的反馈可以采用按规定时间间隔播放某个消息的形式。例如“Please wait while I get your information”（在我接收你的信息时请等待）。

你可以使用下面几种方法来减少用户对出错消息和错误恢复例程的失望感：

- 使用非恐吓性语言来建立出错消息。例如你可以使用这样的出错消息：“Sorry, I didn't understand your input”（对不起，我不明白你输入的信息）（后随原来的提示，或者使用与上下文相关的帮助信息，后随提示的缩写版本）。
- 对出错消息或情况提出一个进行应答的建议。例如，“Sorry, I didn't understand your input. For ‘football’ , press 1, for ‘foosball’ press 2, to start over press star, star.”（对不起，我不明白你输入的信息。如果是“football”，请按1键，如果是“foosball”，请按2键，如果要重来，请按星键）。
- 出现1到2个错误情况后，不要切断与用户的连接。请将他们转到应用程序中的一个适当位置，如果他们不想再试了，他们就会挂断电话。

2. 测试原则4对非约束型应用程序的影响

非约束型应用程序通常拥有平台的支持，能够比约束型应用程序更加有效地为用户提供反馈信息和便于使用的错误恢复功能。非约束型应用程序能够按照下面的方法来处理反馈和出现的错误。

- 在任何应用程序等待来自平台翻译器的应答的时候为用户提供的反馈信息。这个反馈信息可以是一个消息、一个广告，也可以是一个表示“我仍在工作”的声音，或者是给目标用户播放的音乐。

- 当系统发觉与因特网有关的时延可能会造成长时间等待结果时，为用户提供不同的反馈信息。这里出现的问题是，因特网系统的计时器可能要花费很长的时间来提供问题说明。如果应用程序和平台能够发现这个问题，你可以让你的用户知道这个问题，并且使他们能够转到你的应用程序中的另一个模块，开始新的搜索或进行其他操作，而不是切断连接。
- 使用比较复杂的错误处理程序，根据用户在应用程序中所处的位置和依次遇到的错误情况的数量，采用分层次和定制的反馈信息。

5.3.6 测试原则5：保持导航和术语的一致性

在VUI内建立统一的标准，对于你的用户建立应用程序的良好的思维模式来说是很重要的。用户要求你建立的应用程序应该做到使用统一的术语，并且在应用程序的相似模块中使用统一的应答。

将测试原则5用于VUI的设计

应用程序中的一致性有助于给用户造成一种自己能够很好地把握应用程序的感觉。当用户感到不能很好地把握应用程序的时候（即使这种感觉没有什么根据），他们失望的程度就会增加。当用户的失望情绪太严重的时候，你就应该知道将会发生什么情况——他们会挂断电话。

测试原则5对约束型或非约束型应用程序的影响

对于约束型和非约束型这两种应用程序来说，它们对一致性的要求基本上是相同的你可以使用下面几种方法，将一致性纳入你的应用程序：

- 始终对全球导航使用相同的命令。例如，不要在一个模块中使用**来进行返回，而在另一个模块中使用*B来返回。
- 测试，测试，再测试，确保全球导航命令能够在他们需要起作用的地方发挥作用。
- 当在多个模块中使用相同的功能时，本地导航采用相同的命令和提示结构。例如，不要在你的一个应用程序的一个部分中说“*To save your recording, say ‘save’*”（若要保留你的记录，请说‘保留’），而在该应用程序的另一个部分中又说“*To save your file say ‘store’ or press *S*”（若要保留你文件，请说‘存储’或按*S键）。
- 如果你在应用程序的一个模块中提供了DTMF选项作为口语输入的替代方式，那么在该应用程序中的所有模块中都要统一使用这种替代方式。
- 只要可能，应该在整个应用程序中使用统一的声音。

5.3.7 测试原则6：对选择的声音和声音的个性进行评估

现在已形成了对VoiceXML应用程序的用户激烈争夺的态势。选择恰当的应用程序的声音和声音个性，有助于将你的应用程序与其他的应用程序区别开来，并且使你的应用程序能够更好地吸引目标用户。

对声音和声音的个性的选择所包括下面几个方面：

- 语言——你的应用程序支持什么语言？它支持多种语言吗？
- 性别——你的用户是喜欢男性的声音还是女性的声音？
- 音调——你的用户是喜欢音调较高的声音还是音调较低的声音？

- 个性——你的用户是属于普通人群中的一员，还是少数民族或文化团体的成员？是否喜欢使用重复说话的方式、术语或俚语？
- 技术——你的应用程序是使用录制的消息、TTS声音、还是两者混用。

你将如何获得对这些问题的答案呢？在开始进行应用程序的开发时，请将这些答案纳入你的用户分析中。

将测试原则6用于VUI的设计

了解你的用户喜欢什么样的声音和声音的个性，这对于执行用户分析任务来说是非常重要的一项工作，应该在开始进行应用程序的开发时进行。当你得到这些问题的答案后，你就可以用他们来选择声音，并且指导系统提示及应答的设计。

1. 测试原则6对约束型应用程序的影响

选择恰当的声音和声音的个性，对于约束型应用程序来说并不是十分关键的一项工作，因为通常没有许多选择工作要做。对于约束型应用程序来说，声音的选择主要是为录制的通告来选择声音。而对于TTS的声音或者声音的个性而言，可供约束型应用程序选择的选项范围往往非常狭窄。

若要解决约束型应用程序的声音和声音的个性问题，你可以使用下面几种方法：

- 挑选一种你的大多数目标用户都接受的录制通告的声音。例如，对北美用户的研究显示，一种女低音的使用可能是绝大多数人都能够接受的。
- 针对用户的提示和消息的设计要符合目标人群、民族或文化团体的需要。约束型应用程序针对的目标用户往往非常广泛，在这种情况下，提示应该力求简单，不要使用拟人化的语言，也不应该使用像“Please”、“sir”之类不必要的客套话。
- 选择尽可能接近录制通告声音的一种TTS声音，至少这个声音应当在性别上相吻合。不一致的声音会增加用户的负担，你知道其结果是用户挂断电话。

2. 测试原则6对非约束型应用程序的影响

选择恰当的声音和声音的个性对于非约束型应用程序来说是非常重要的，因为它们提供了一些极好的机会，以便为最合适的用户群体定制应用程序。声音和声音的个性正被广泛应用于使VoiceXML服务做到与众不同。

如果没有良好的用户数据，你就无法为非约束型应用程序很好地决定究竟选择什么样的声音和声音的个性。在你作出初步的决定后，要在你的目标用户群体中对你选择的声音和声音的个性进行检验，并且根据需要对它们进行修改。你为吸引特定用户群体而设计的提示，应该使用第6章中介绍的技术，在开发周期的早期进行测试。

若要解决约束型应用程序中的声音和声音的个性问题，你可以使用下面几种方法：

- 选择一个符合用户要求和希望的录制通告的声音或TTS声音。对于某些平台来说，必须使用录制的声音。其他平台则可选用能够接受的TTS声音。
- 如果平台的质量确实是非常逼真的，那么请选择一种TTS声音。如果所有内容都可以用TTS来传递，那么不管是传递系统提示，还是传递因特网内容，用户就会感到应用程序显得与他们亲密无间。
- 在目标用户中测试选择的声音。如果用户不能接受TTS声音，可以转用录制通告的声音。就VoiceXML应用程序的开发来说，它的一个好处是你总是要编写提示的，所以你可以很

容易地为录制的通告声音建立脚本。

- 使用目标用户的术语、俚语和说话方式来设计提示，在开发过程的早期就对提示进行测试。

5.4 UCD和系统设计的参考资料

你可以使用许多著作和Web资源来帮助你提高VUI的设计技能，这些资源包括：

- Ameritech Phone-based User Interface Standards and Design Guideline（基于Ameritech电话的用户界面标准和设计指导原则），Schwartz与Hadzinski合著，“Human Factors”，1995年出版，书号37(2) 251-264。
- Designing Effective Speech Interfaces（设计有效的语音界面），Weinschenk和Barker合著，书号ISBN：0-471-37545-4。
- Human Factors and Voice Interactive Systems（人的因素与语音交互式系统），Daryle Gardner-Bonneau编辑，书号ISBN: 0-7923-8467-9。
- International Telecommunication Union Recommendation F.902，“Human Factors-Interactive Services Design Guidelines”（国际电信联盟建议F.902，“人的因素——交互式服务程序设计的指导原则）。
- ISO 13714，“User Interface to Telephone-based Services-Voice Messaging Systems”（用于基于电话的语音界面—语音传信系统）。
- Tellme Studio for VoiceXML developers（VoiceXML开发人员的Tellme工作室），网址是<http://studio.tellme.com/>。
- The Usability Engineering Lifecycle（可用性工程设计的寿命），Deborah Mayhew著，书号ISBN:1-55860-561-4。
- Usability Engineering（可用性工程设计），Jakob Nielsen著，书号ISBN:0-12-518406-9。
- Usability Inspection Methods（可用性测试的方法），Nielsen和Mack著，书号ISBN:0-471-01877-5。
- User and Task Analysis for Interface Design（用于界面设计的用户分析和任务分析），Hackos和Redish著，书号ISBN:0-471-17831-4。
- VoiceGenie developer support（VoiceGenie开发商的支持特性），网址是<http://www.voicegenie.com/index.html>。

5.5 小结

本章对采用通用UCD过程的VUI设计进行了全面的介绍。UCD将重点放在产品的设计上，并且从用户、用户的任务和用户界面的角度来进行产品的开发，从而提供了一种开发符合用户需要的应用程序的方法。

本章的内容从应用程序的开发过程转到了开发过程的一个组成部分，即VoiceXML可用性测试。所谓测试原则，就是进行用户界面的评估时通常使用的指导原则。在这里，通过介绍“逆向设计”的测试原则，可以了解你应该如何将这些测试原则用作应用程序设计的指导原则。

下一章将要介绍如何使用本章提供的设计指导原则来开发应用程序的原型。

第6章 VUI应用程序原型和可用性测试

前面几章介绍了用VoiceXML建立的语音用户界面和VUI设计的概述。本章将要提供一些具体的说明和例子，以便展示究竟如何建立应用程序的原型，并且评价和测试不同完整度的应用程序的可用性。

注意应用程序的设计、原型的建立、评价和测试等问题，有助于确保应用能够便于使用，并且使你的用户能够通过电话来完成他们的工作。它还有助于确保你能够在用难以修改的代码编写应用程序之前，尽早发现应用程序的可用性和产品设计方面存在的问题。

本章将要介绍下列几种完整度的应用程序的建立和可用性的评估和测试：

- 应用程序的设计方块图和流程图。
- VUI脚本——应用程序的测试版。
- VoiceXML代码的原型。

根据我们了解的我们与之共同工作的小组的需要，本章介绍的VUI应用程序原型的详细情况的重点将放在方块图和VUI脚本程序上，而可用性测试的细节将重点介绍VoiceXML代码的原型。

6.1 UCD的前端工作

本节将要介绍在第5章中讲述的以用户为中心的设计（UCD）过程的早期阶段中应该做的前端工作：

- 第一阶段——用户分析和任务分析。
- 第二阶段——任务环境、应用程序结构的含义、可用性目标和低保真度VUI原型。

要想开发一个出色的VoiceXML应用程序，从某种程度上取决于是否准确地确定了应用程序的目标用户群体。你常常是脑子里先有一个设想，然后再回过头来确定潜在的用户群体。这种情况不一定不好，只要你不是等到对应用程序进行编码的时候再来确定谁是你的用户，他们希望你的应用程序具备什么功能，以及你开发这个应用程序究竟是否有市场。

因此，本书中的示例程序将类似于你曾经使用过的应用程序，我们决定使用VoiceXML访问的负责销售图书和T恤衫的在线商店这个应用程序例子，因为这正是我们要做的事情。我们有人为该应用程序进行高水平的设计和代码模块的开发，但是，我们没有费多大劲儿，就了解到关于我们的用户及其任务的详细情况。

收集和分析用户数据

你将要建立一个示例应用程序，用于在线销售VoiceXML图书和T恤衫。你可以使用许多种方法来搜集用户数据，以便用于市场分析，搜集数据的方法有的花费很大，有的则几乎是免费

的。比如，你可以

- 向市场调研公司购买数据。
- 对现有的客户群（如果有的话）进行情况调查，或者向他们发放问题调查表。
- 对来自因特网上我们其他的电子商务或移动商务站点（如果有的话）的数据进行分析。
- 对重点人群展开调查。
- 搜索因特网上的免费数据。
- 对同事、朋友和家庭进行采访。

快速搜索因特网，可以获得关于在线购买非小说类图书的用户的（未经核实的）一般特征：

- 大多数是女性，年龄为24岁或24岁以上。
- 收入一般在5万美元以上。
- 占图书总销售额的10%左右。
- 能够访问因特网，并且定期使用因特网。
- 拥有移动电话，并且定期使用它。
- 比较有可能购买精装本图书和大部头平装本图书。

不错，这看起来对我们相当有利。现在让我们来想一想通过VoiceXML应用程序购买图书和T恤衫的用户群体与一般市场数据中所说的用户群体之间有些什么不同。如果你认为用户群体中可能包含了一些“新新人类”（alpha geeks），并且这种人的人数不如“普通人”那么多，那么请给你自己一颗金星。技术图书和T恤衫的市场不如史蒂文·金的最新惊险小说市场那么大。从某种意义上说，这是通过在初期数据搜集时确定非小说类图书购买者而了解到的情况。不过你仍然需要更好地掌握你的用户群体都是些什么人，他们是否或多或少喜欢通过VoiceXML应用程序来购买图书。如果用户群体的规模太小，那么应该考虑开发别的应用程序。

<p>Project Name VoiceXML Book and T-shirt M-commerce Application</p> <p>Project Summary A VoiceXML application for selling "Voice Application Development with VoiceXML" books and T-shirts.</p> <p>Primary User Group Early users and developers of new technology products, often referred to as "alpha geeks."</p> <p>Secondary User Groups All groups that purchase non-fiction books, including technical hobbyists, learning institutions, and corporate training departments.</p> <p>Primary User Group Characteristics General characteristics of the target group for the application include: <ul style="list-style-type: none"> • Male, 24 years of age or older, income above \$50,000 (US). • Account for about 10% of all book sales, preferring hardback books and large paperbacks. • Use Internet, cell phone, electronic organizers, and hand held computing devices. • Like to advertise their advanced adoption of technology (geekiness) through T-shirts and other accessories. </p> <p>Implications for Application Design The primary user group: <ul style="list-style-type: none"> • Is usually in a hurry, so the application should streamline the purchasing process. Use direct, brief prompts. • Is not defined enough to specify a particular voice or speech style. Use a female contralto voice for initial prompts and test with target users. • Often call from cellular telephones in noisy environments, so highly accurate speech recognition is important. DTMF alternatives for spoken input are indicated. Small grammars are indicated. </p> <p>Other User-related Issues The user analysis identified these user-related issues: <ul style="list-style-type: none"> • Corporate or institutional bulk purchases should be supported. • Evaluate offering the book in an electronic format that can be downloaded into the most popular handheld units. • Evaluate expanding the application to offer other VoiceXML accessories such as coffee cups, mouse pads, and baseball caps. </p>

图6-1 用户分析报告的举例

对Web作进一步的快速搜索，你会得到下面这些用户数据，这些数据虽然不完整，但是能够支持影响应用程序开发的思路：

- “新新人类”喜欢购买说明其明显怪癖的东西（比如T恤衫）。
- IT市场继续扩大，但速度放慢了。因此应该为我们的VoiceXML图书建立一个培训市场。
- 电子图书在“新新人类”中非常流行，但是他们抱怨缺乏用于他们的手机的参考书目。
- 企业正在寻找降低IT培训费用的方法，这是我们的图书能够填补的一个空白。

在现实环境中，这时你可能会购买相应的用户数据，以便更好地确定购买你的产品的人数。你也可以继续开发该应用程序，如果应用程序的开发和托管成本较低的话，更加可以如此。在这种情况下，你将继续进行应用程序的开发工作。

图6-1显示了根据上述数据制作的一个用户分析报告。

6.2 应用程序的任务分析

第5章讲述了任务分析对应用程序设计的重要性，因此在这个应用程序开发的例子中我们也不能跳过这个步骤。在任务分析期间，应用程序设计人员要分析用户今天做某件事情时（这里是指购买图书和T恤衫）的工作过程。然后你要确定你在应用程序中必须支持哪些任务，以及如何用你的应用程序使这些任务实现自动运行或者将它们激活。

6.2.1 新技术应用程序开发中的任务分析

在继续下面的内容讲解前，先让我们澄清一些进行新技术应用程序开发的任务分析时的常见错误观念。

有些人认为，进行新技术应用程序开发时，你无法进行任务分析，因为目前还不存在新技术应用程序的用户。这些人忽略了这样一个问题，即提出全新的任务通常并不是新技术或新技术项目的结果。一般来说，从用户角度出发，采用新技术的目的是为了提供一个更容易和更有效的方法，以便从事他们已经用一种或另一种方式进行的工作。

我们认为，地球上不存在完全是新的任务，因此你总是可以通过分析与当前工作关系最密切的例子，学会如何设计最有效的应用程序。

6.2.2 任务分析的举例

你可以使用若干种方法来确定当前使用的任务进程，这些方法包括：

- 个人对任务的了解。
- 重点用户群。
- 调查。
- 观察执行任务的人。

在应用程序开发的这个阶段过分依赖于个人对任务的了解，这是危险的，原因是多方面的。你可能不是目标用户群的成员，因此你执行任务的方法与目标用户组群的代表所用的方法不同。你可能并不真正懂得如何正确或有效地执行任务。你的工作环境和资源不同于目标用户的工作环境和资源。所有这些因素都会导致你采用不太好的设计思路。当设计人员采用不太好的设计

思路之后，要到应用程序开发完成时才能发现设计中存在的错误。

根据我们的经验，如果你混合采用多种不同的方法，那么就更有可能取得成功。例如，如果你个人知道任务的范围，那么你就能够完成下面这个由3部分组成的任务分析：

- 1) 根据你的经验确定任务并且画出工作流程图。
- 2) 使用重点用户组或通过调查，确定你遗漏或理解不正确的工作过程。
- 3) 观察或者采访目标用户组，以便最后确定工作过程的细节。

根据本书的需要，假设我们采用这个由3部分组成的任务分析过程，那么这个分析将要回答这样一个问题，即“人们如何购买图书和T恤衫？”

假设这个分析确定了用户可能执行下面这些任务：

- 根据作者来搜索图书。
- 搜索感兴趣题材的图书。
- 浏览图书或T恤衫，根据题材、出版日期、特殊兴趣的用户组和图形等因素，了解可以买到什么样的图书。
- 评估图书或T恤衫的物理特征，比如颜色、纹理和重量等。
- 检查T恤衫是否合身。
- 检查T恤衫的图形质量。
- 与朋友或专家探讨购买的可能性。
- 选购最低价格的商品并了解是否有货。
- 实施购买。
- 取消购买。
- 退回购买的商品，并获取退款。

作为任务分析过程的一部分，通常你应该确定与该任务相关的工作流程。图6-2说明了购买图书和T恤衫的部分工作流程。

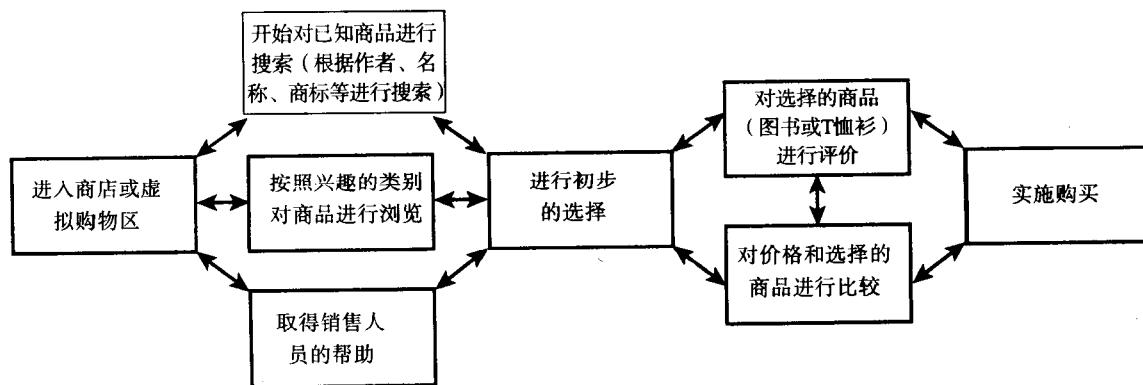


图6-2 当前工作流程图用于确定应用程序设计的时机

6.3 早期的应用程序原型——应用程序流程图

应用程序的基础是根据任务分析来重新设计的工作流程。这个重新设计的工作流程能够显

示你在应用程序中打算支持哪些用户任务，以及这些任务之间的关系是什么。

根据本书的需要，我们在示例应用程序中将不考虑下面两类任务：

- 需要与你打算销售的产品进行实际的交互操作的任务。
- 需要搜索或浏览的任务。请记住，该示例应用程序只用于销售VoiceXML图书和T恤衫。

该应用程序运行的起点将包括执行下列核心任务：

- 将商品添加给购物车。
- 从购物车中删除商品。
- 审查购物车里的商品及价格。
- 连接到操作员（销售助理）。
- 完成商品的采购。
- 添加审查的说明。
- 收听审查的说明。

6.3.1 建立低保真度的应用程序原型

工作流程图或使用实例可以建立很好的低保真度VoiceXML应用程序的原型。我们选用了工作流程图，因为它们代表了用户和开发人员都能理解的应用程序。

图6-3显示了你可以用来说明应用程序的设计和评估应用程序可用性的重新设计的工作流程图。

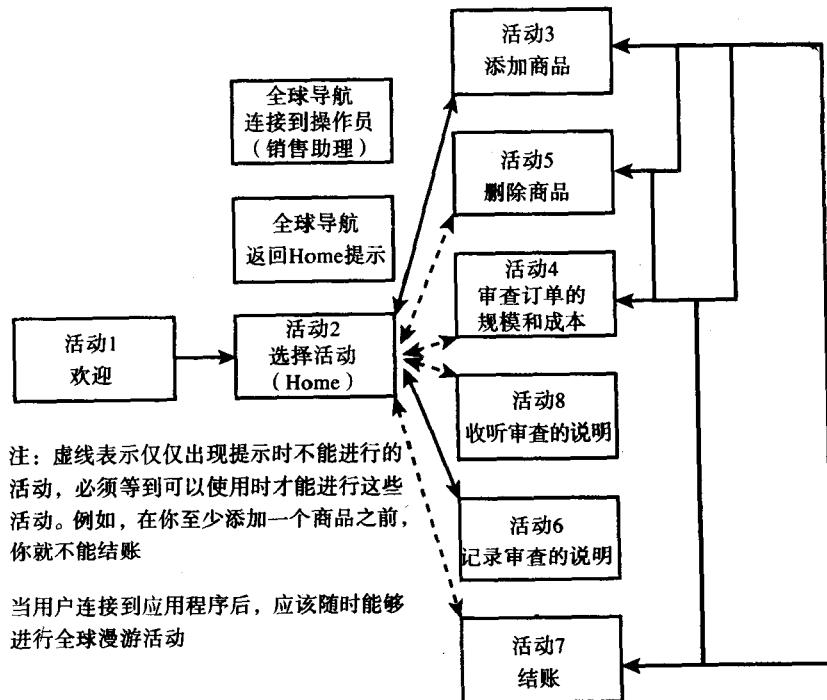


图6-3 这个简单的流程图提供一个模式，用于说明
应用程序的情况和进行早期的可用性评估

6.3.2 用流程图来说明应用程序的功能

第5章介绍了一个以用户为中心的应用程序设计过程，并且提到了许多不同的输出可以用于与应用程序开发人员进行信息的交流，检验开发工作进行的情况是否符合应用程序设计的要求。

即使在应用程序开发过程中的这个早期阶段，这个流程图也清楚地展示了（这个设计阶段）该应用程序支持哪些任务的执行，即购买T恤衫、购买图书、更改订单、了解订购的商品的费用、连接到操作员等。另外，它还暗示了该应用程序不支持哪些任务，比如T恤衫纹理的感觉，图书纸张的重量，T恤衫的试穿，浏览图书的内容等。你可以使用这个流程图来探讨为何包含或不包含其他任务的原因，因为你可以了解某个任务模块与其他模块之间有些什么关系，并且能够对它给VUI增加多少复杂性的问题有一个初步的概念。

6.3.3 根据流程图对应用程序的可用性进行评估

你可能记得，第5章我们介绍过应用程序的可用性测试和可用性评估的问题。应用程序的可用性评估的方法是，对应用程序与一组可用性测试原则进行比较，并且注意它们是否一致。可用性测试的方法是让一群目标用户试用应用程序的原型、测试版或已经安装运行的版本，以便执行应用程序支持的工作。

图6-3中的流程图太简单了，因此无法用来进行更多的可用性测试，不过它已经足够可以用来进行可用性评估，并且可以用来与真正的用户讨论应用程序的有关问题。你可以通过这个流程图来了解应用程序是否遗漏了某些重要的功能模块，或者在继续进行应用程序的设计时，是否需要解决某些明显的导航问题或系统结构问题。

通过VUI流程图进行应用程序的可用性评估

下面我们要根据第5章中介绍的VUI应用程序可用性测试原则，对图6-3中的流程图进行应用程序的可用性评估。如果你没有阅读第5章的内容，那么你应该复习一下表5-2的内容。可用性测试原则包括下面几项：

- 尽量降低对用户的记忆和识别过程的要求。
- 保持简单而自然的对话框。
- 提供清楚的导航和快捷操作方式。
- 提供一致的反馈信息和便于使用的错误处理工具。
- 保持导航和术语的一致性。
- 对选择的声音和声音的个性进行评估。

表6-1按编号列出了各个可用性测试原则，并且列出了可用性评估的说明。在实际应用环境中，你可以使用一个比较表格，来记录你的评估情况，抓取你发现的问题，供你将来更加详细地进行应用程序设计时使用。请注意，该表列出了可能的解决方案，但是不要在应用程序开发过程中这样早的时候就过多地去考虑如何解决应用程序设计中出现的问题，这一点很重要。在这个阶段中，你应该抓住对代码编写影响较大的应用程序的可用性问题，而不是去完成每一项详细的设计。

表6-1 应用程序可用性评估流程图举例的说明

编号	评估说明	可能的解决方案
1	这个时候对第一个可用性测试原则还很难说。活动4、6和7可能会变得复杂起来	现在可以将这些活动标为可能存在可用性问题，然后进行下一步分析
2	这个时候很难进行评估	当你以后更加详细地建立对话框时，应该对这个指导原则进行重新评估
3	本流程图显示了从欢迎活动开始的比较简单的导航方式	可以考虑添加全球导航元素，比如“help”（帮助），“start over”（重来）或“check out now”（现在结账）
4	本流程图中不包括错误处理。在这种情况下，我们只是尚未达到那样详细的工作程度。但是该流程图中列出的各种活动有可能产生错误条件	记下潜在问题的范围。当你继续进行应用程序的设计时，应该尽可能避免产生错误条件，并且在其他所有地方提供适当的错误恢复功能
5	有些术语与用户的思维模式可能不一致。可能不一致的地方之一是在“连接到操作员”模块中	你应该设法建立一个采购和购物是思维模式“与销售助理交谈”可能更合乎逻辑
6	你可以从流程图中听到声音或个性特征	请根据用户分析提供的声音和提示的指导原则。可以过后再进行测试，但是越早越好

根据流程图通过用户进行应用程序的可用性评估

虽然这时流程图还不够详细，无法进行应用程序的可用性测试，但是它已经具有足够的详细度，可以供目标用户群中的人对应用程序进行评估。如果你设计的第一个应用程序原型没有最终用户输入的信息（正如本例中的情况那样），那么现在由最终用户来评估应用程序的可用性，是个很好的想法。

这个阶段的用户评估可以采用下面的方式来进行：

- 讨论——向用户展示流程图，征求用户对流程图的看法，请用户提出修改建议。
- 建立重点用户组——将一组用户集中起来，请他们展开讨论，以便了解他们希望应用程序具备什么功能。向用户组展示流程图，征求他们的意见，了解他们使用应用程序的频度。
- 举行自由讨论会——举行自由讨论会，以便了解原始应用程序方案中可以包含哪些补充特性，或者这些补充特性是否可以纳入其他应用程序。

这时你获得了用户分析和任务分析提供的数据，以及用户对你的低保真度的流程图原型的反馈意见。下一节将要介绍一些方法，以便在投入大量精力进行代码编写之前，对应用程序的设计进行改进和调整。

6.3.4 应用程序可用性要求举例

当用户分析和任务分析已经完成，应用程序已经进行了初步的定义之后，下一步就应该确立应用程序的可用性要求和目标。可用性要求和目标常常与应用程序的要求或设计文档中为软件产品书写的其他类型要求结合起来。

表6-2列出了应用程序可能的应用和用户感觉的要求和目标。最后选定的目标和要求通常是

在要求文档进行技术审查时完成的。

表6-2 建议的应用程序的可用性目标和要求

目标或要求	说 明
用户应该在他与服务程序建立连接后的1分钟内完成一本书的购买	目标时间是根据移动电话呼叫和对速度的要求来确定的
用户应该能够在他与服务程序连接后的1分半钟内完成一本书和一件T恤衫的购买	这些目标可以实现一个简单的设计方案，并且建立可用性测试时使用的参数
当系统运行繁忙时，用户可以在对提示作出应答后1.5秒钟内听到“系统繁忙”的声音	用户必须知道系统仍然在工作，否则他们可以关机
当外部系统没有作出响应时，应用程序便在8秒钟内向用户提供反馈信息。用户可以选择切断连接，也可以继续等待	所有这些时间都是语音应用程序标准建议的因特网时间的修改版本
在2秒钟内输入信息后，为用户提供反馈信息，方法是转入下一个模块，或者传递错误恢复信息	这些较短的时间应该适用于高效率应用程序。如有必要，应该进行测试和修改
应用程序无输入的定时器应该设置为成功地传递要求输入的提示之后3秒钟内到期	
进行可用性测试的人将把应用程序的使用方便性定为5个刻度的标尺上的3.75或更好，标尺上的刻度5表示“使用起来非常容易”	通过调查来衡量进行测试的人。如果这个测试目标失败了。请对有关位置上的可用性问题进行分析，以提高应用程序的可用性
所有进行可用性测试的人中至少75%的人能够在他们初次试用该应用程序时成功地购买图书，并且不需要进行任何形式的应用培训	应该确保人们能够调用该应用程序，并且能够进行购物。他们将不会花时间去做其他任何事情
所有进行可用性测试的人中至少60%的人将该应用程序的声音可懂度定为5个刻度标尺上的3.5或更好，标尺的刻度5表示“非常容易懂”	人们必须能够理解应用程序的声音，不管是录音还是文语转换的声音。一种相关的应用程序测试要求可以写成“pleasant to listen to”（很好听）

建议的其他应用程序可用性要求和目标包括：

- 正确识别符合语法的语句应该达到的百分比。
- 不能正确识别不符合语法的语句应该达到的百分比。
- 正确拒绝不符合语法的语句应该达到的百分比。

6.3.5 初始语法设计

用户分析和任务分析为VoiceXML应用程序开发的一个重要组成部分提供了输入信息，这个组成部分是许多人所不太熟悉的，这就是应用程序的语法设计。你希望应用程序识别的词汇或词组必须包含在语法中。这听起来似乎很简单，但是实际上并不简单，它关系到应用程序设计的成败。

第5章讲述了几种方法，说明你可以使用语法来支持环境的、平台的或者应用程序的限制条件。下面让我们根据本章中的示例应用程序更加具体地说明这个问题。

语法太复杂常常会给应用程序带来许多问题，因为它会导致识别错误。

语法太简单则会给VUI带来问题，因为它需要一些规定的提示，以便实现类似老式交互式语音应答系统的应用。

表6-3提供了一个如何使用应用程序方块图来开发初步的口语语法和DTMF语法的例子。初始语法文档必须随着应用程序的演变而加以修改。也许你希望对与特定应用程序的活动相关的多个语法上的术语进行区分。

表6-3 初始应用程序语法的举例

应用程序的活动	口语语法的术语	DTMF语法
全球活动—连接到操作员 (销售助理)	必须进行测试，以便从下面几个术语中作出最后选择： “Operator” “Sales Assistant” 其他替代术语 建立选定术语的置换方法	*0 (星号, 0)
全球活动—返回“Home” 提示	必须进行测试，以便从下面几个术语中作出最后选择： “Home” “Go Home” “Start Over”	** (星号, 星号)
全球活动—获取帮助 添加商品	“Help” “book” “tee shirt”	1 2
结账	美国的所有州名称	

当大多数语法术语确定之后，应该对语法进行评估，以了解哪些发音非常相近，从而导致识别器无法将它们区分开来。下面是可以使用的几种解决办法：

- 使用多种语法。
- 选用对用户来说含义相同的替代术语。
- 添加前缀词，表示这是个命令，并且将它与声音相似的术语区分开来。

关于DTMF语法的其他帮助信息

由于许多人都具备使用传统的专用交互式语音应答(IVR)系统的经验，因此只要适合的时候，就可以将这些系统的原则用于DTMF语法。

关于DTMF的指导原则的补充信息，请参见：

- ISO 13714，“User Interface to Telephone-Based Services-Voice Messaging Systems”(与电话服务系统——语音传信系统之间的用户界面)(<http://www.iso.ch/>)
- 国际电信联盟建议F.902，“Human Factors-Interactive Services Design Guidelines”(人的因素——交互式服务程序设计的指导原则)(<http://www.standards.com.au/>)
- 澳大利亚/新西兰标准AS/NZS 4263:1997(<http://www.standards.com.au/>)

6.3.6 与开发人员交流信息的结果

如果要成功地开发应用程序，系统工程师、软件开发人员和软件测试人员应该从一开始就对应用程序的设计方案提供输入信息。

如果开始时做不到，那么应该在应用程序的开发过程中提供这样的输入信息。应该将所有输入信息综合起来，以便运用于整个设计过程，并且不断取得补充的输入信息，着手发现可能影响应用程序设计的实现问题。

他们的输入信息可以用于下列目的：

- 改进应用程序的设计。
- 添加关于错误检查、错误恢复和帮助例程的高级说明。
- 添加关于数据库访问例程的高级说明。

你应该建立一个更加详细的高水平应用程序原型，请开发人员和典型用户对它进行审查。

这时，你应该给应用程序制作一幅比较详细的流程图，即通常我们需要的VoiceXML应用程序流程图，然后建立该应用程序的脚本程序版本。

6.4 VUI脚本——中间阶段的应用程序原型

VUI脚本是个应用程序设计文档，它包含应用程序的详细文本版本。VUI脚本包含下列要素：

- 应用程序语法的词语（口语和DTMF）。
- 应用程序行为特性的指导原则。
- 应用程序的提示、帮助信息、出错消息等使用的实际文本。
- 当用户对提示作出响应时，显示用户应该转向何处的转移选项。

你可以使用流程图工具和文字处理程序来建立VUI脚本，将方块图从流程图工具输入到文字处理程序。

根据我们开发VoiceXML应用程序的经验，我们发现VUI脚本是以用户为中心的设计对应用程序开发过程最有用和功能最强大的工具。VUI脚本不需要编写一行代码，然而它却能够起到下列作用：

- 提供一个清楚和明确定义的应用程序模型。
- 可以相当有效地进行应用程序的可用性测试。
- 为软件开发人员提供可以重复使用的数据源。
- 为确定定时问题，确定初始定时设置值和高层资源要求提供一个基础。

通常而言，你是按照逐个活动来建立VUI脚本的，然后将详细信息添加给前面介绍的VUI方块图的最后一个替代活动。结果将产生一个更加详细的方块图与详细提示、帮助文本、出错消息文本和逻辑转移选项的组合。

这个组合可以非常有效地成为开发人员的工作和进行可用性测试的基础。

6.4.1 可用性测试举例

你可以用多种方法将VUI脚本用于应用程序的可用性测试。表6-4汇总列出了使用VUI脚本

作为应用程序原型进行可用性测试的方法举例。如果你需要进一步了解可用性测试的详细说明，请阅读雅葛布·尼尔森撰写的《Usability Engineering》(可用性工程设计)和雅葛布·尼尔森与罗伯特·麦克合编的《Usability Inspection Methods》(可用性检查的方法)这两本书。

表6-4 用VUI脚本进行应用程序的可用性测试

方 法	说 明
按测试条件进行评估	<p>聘请某个以前从事过语音应用程序开发的人，请他使用第5章介绍的测试原则，以VUI脚本的形式，对你的应用程序进行可用性评估</p> <p>根据需要修改VUI脚本</p>
测试应用程序的输出消息	<p>聘请一名测试人员，此人并不熟悉你的应用程序，但是他是你的目标用户。如果能够得到帮助，那么当你进行测试时，可以请某个人给VUI脚本的拷贝做上标记</p> <p>阅读给测试人员的每个提示、帮助消息和出错消息。当消息文本是口述的而不是默念的时，确定消息文本是否可以听得懂</p> <p>你并不是要测试应用程序的逻辑方法是否正确，你只是测试应用程序给VUI的静态输出</p> <p>根据需要更新VUI脚本，如果发现你的应用程序问题很大，那么你要用很长时间才能从失望中恢复过来</p>
通过试运行来发现问题	<p>聘请你的一群同事</p> <p>向这些同事全面介绍你的应用程序要实现的功能和目标用户组的特点</p> <p>让你的同事遍历应用程序流程图，使他们了解应用程序运行过程中的每个重要活动，并且阅读用户将能听到的各个提示</p> <p>请他们提出问题，使他们了解用户如何与应用程序进行交互操作，并且请他们就如何改进应用程序的问题提出建议</p>
根据条件进行可用性测试	<p>建立一些可用性测试的条件，这些条件要包括执行你的应用程序的重要任务的用户</p> <p>聘请一些测试人员，他们没有接触过你的应用程序，但是又属于目标用户组的成员</p> <p>聘请一位助手，当每个可用性测试条件运行时，要进行详细的记录</p> <p>运行每个测试条件。你将作为应用程序。你将读取测试条件显示的提示，然后等待测试用户作出的反应。你的助手则负责记录运行每个测试条件时出现的可用性问题。如果能够得到相应的资源，可以制作测试情况的录音带或录像带，供将来参考</p> <p>对测试条件运行过程中发现的可用性问题进行分析，根据需要修改VUI脚本</p> <p>由新的测试人员来运行所有受影响的测试条件的另一个替代版本</p>

注意 你为测试VUI脚本原型建立的可用性测试条件可以反复用于测试应用程序的工作原型或测试版应用程序。它们只需要修改一下，就可以访问因特网数据和动态内容。

6.4.2 可重复使用的资源

应用程序开发小组常常发现VUI脚本是非常出色的资源。当开发人员拥有VUI脚本这样的资源时，他们就可以集中力量编写使应用程序能够运行所需要的代码。

由于VUI脚本包含的已经经过测试和拼写检查包含的，因此它们可以在开发人员选定的VoiceXML标记内直接进行剪切和粘贴。

VUI脚本包含大量关于应用程序逻辑的信息，它为开发人员提供了应用程序应该如何运行的清楚概念，从而可以减少因为开发人员猜测应用程序的设计应该如何用代码来加以实现而进行软件开发所需要投入的精力，并且以后也不需要进行返工。

对脚本进行录音

如果你用专业技巧对提示、帮助和出错消息进行录音，那么VUI脚本可以用另一种方式进行重复使用。由于这些消息在VUI脚本中做了最后定型，因此它们可以转交给录音工作室去进行录音处理。

当录音工作室从事它的录音工作时，你可以使用VUI脚本，用非专业人员的声音对照它的草稿进行录音，将录音由于应用程序的工作原型。使用相同的文件名，可以在提交录音时很容易将非专业录音转换成专业录音。

6.4.3 定时问题与资源要求

由于VUI脚本是一个详细的不能实际运行的应用程序原型，因此它能够为你提供确定定时问题和核实对资源要求的初步估计时需要的大量信息。

表6-5列出了你用VUI来确定和解决定时问题和资源问题时采用的一些方法。

表6-5 使用VUI来满足定时和资源需求

当你需要…	那么你可以…
确定用户成功地与应用程序保持连接的最短时间	<p>使用你的“Purchase a book”（购买图书）应用程序可用性测试条件，确定通过应用程序的最佳路径。读取每个系统输出，记录每个系统输出的时间。模仿用户输出，记录每个用户输出的时间。统计交互式操作的数量，添加每个交互操作估计的系统等待时间</p> <p>将所有时间加在一起，得出成功地运行你的应用程序所需要的最小时间的准确估算值</p> <p>对时间值进行分析。了解你的目标用户群是否能够接受这个时间值。如果不能接受，请寻找相应的办法缩短该时间值</p> <p>使用上面确定的系统输出时间</p> <p>选定一个你认为你的用户群收听提示时能够接受的时间值，比如5~8秒钟</p> <p>对超出你选定的时间限制的提示进行分析。它们能否缩短？你能否添加一个快捷方式，使有经验的用户能够避开这个提示。由用户进行测试，以了解提示的内容是否长得无法接受</p>
评估应用程序的“厌倦因素”	

(续)

当你需要…	那么你可以…
评估平台资源的需求	(根据你以前建立的可用性测试条件)为输入和输出评估关键任务通过系统的路径 确定每个路径有多少百分比用于输出TTS,输出录音通知,收集口语输入信息,以及收集DTMF输入信息等 这些结果与估算的对平台的输入呼叫负载相结合,将有助于确定平台组件的大小

6.5 VoiceXML代码原型

如果你根据上一节建议的方法,设计、开发和测试了一个详细的VUI脚本,那么用VoiceXML开发应用程序的工作应该能够迅速地进行。你的下一个输出将是一个差不多能够运行的应用程序原型。

你可以在活动这一层次上建立VoiceXML应用程序的原型,你也可以建立一个完整的应用程序原型。应用程序原型也许无法访问因特网上的实际内容。它可能不具备用专业声音录制的消息。

你想要在什么层次上建立应用程序的原型,都是可以的,只要应用程序原型是在你的测试平台上运行。你的测试平台可以是因特网上的一个VoiceXML开发商的站点,也可以是在你的个人计算机上运行的专用软件开发商的软件系统,或者是你工作中使用的VoiceXML翻译程序平台。

6.5.1 可用性测试的前提条件

下面各节将要介绍对功能有限的应用程序原型进行可用性测试的情况,它的前提是已经开发了一个完整的VoiceXML应用程序,并且已经安装在测试平台上,能够访问真实的或者模拟的因特网数据。换句话说,这个应用程序是如此完整,你可以对它进行所有特性的测试,不过下列特性不能测试:

- 真实环境中的系统等待时间。
- 专业录音通告的质量。
- 与应用程序为输出给用户使用而检索的因特网内容的长度或格式相关的各种问题。

6.5.2 可用性测试过程的概述

深入介绍对VoiceXML代码原型、代码测试版或已经安装运行的代码进行可用性测试的完整过程,这不属于本书讲解的内容范围。如果你想要了解这方面的详细信息,请参阅雅葛布·尼尔森撰写的《Usability Engineering》、雅葛布·尼尔森与罗伯特·麦克合编的《Usability Inspection Methods》、杜马与雷迪西合著的《A Practical Guide to Usability Testing》(可用性测试实用指南)等著作。

与可用性测试相关的高层次活动包括下列方面:

- 确定用户群的特性，以便选定测试人员，并且确定每个用户群需要的测试人员的数量。
- 确定要测试的任务和功能。
- 确定测试地点的要求。
- 编写有关的材料，用于确认测试人员、版本格式和进行询问式调查。
- 制定可用性测试方案。
- 审查可用性测试。
- 聘请测试人员
- 实施可用性测试。
- 评估测试结果，制定解决方案。

6.5.3 制定可用性测试方案

如果你遵循本章介绍的工作过程，那么你要制定一些简单的可用性测试的方案，以便对你的VUI脚本原型进行可用性测试。这些方案应能重复使用，不过这些方案的详细程度可能达不到对应用程序原型进行测试所需要的水平。

如果你过去接触过可用性测试的工作或人的因素方面的工作。你可能担心你能否设计、建立和进行可用性测试。你可能设想过建立上百万美元的可用性测试实验室，进行有关的测试，以便跟踪用户最小的肌腱开关。虽然有些应用程序需要进行详细的和符合科学的准确的可用性测试，但是，通过使用雅葛布·尼尔森在他的《Usability Engineering》一书中所说的“折扣”可用性管理操作，就可以大大提高大多数应用程序的性能。

为了执行这里所说的可用性测试方案，你必须建立一个“测试实验室”，这是一个不受任何干扰的安静区域，它配有：

- 舒适的座位。
- 能够接触你的测试平台。
- 如果你的测试平台支持拨号接入的话，则配有一个免提电话。如果免提电话降低了你的语音识别器的准确性，那么你必须配备一个替代设备，以便监控应用程序的输出。
- 一台磁带录像机，最好它能够同时记录用户与你的应用程序进行交互操作的双方的情况。
- 一名助理观察员，在测试过程中帮助你记录可用性方面的问题。

本节将要介绍一种低成本的可用性测试方法，它能够暴露用户遇到的大多数问题，这样你就可以在将应用程序提供给用户使用之前将这些问题解决掉。这种可用性测试方法采用一些简单的观察和标记方法，并且可以与下面两个等级的可用性测试方案结合起来使用：

- 高层方案，用于评估用户对应用程序的基本特性和功能的初步反应。
- 低层方案，是一种更加详细的方案，专门用来测试应用程序中对可用性影响很大的各个元素。

1. 高层可用性测试方案

高层可用性测试方案用于评估用户初次接触应用程序及其主要功能时应用程序的可用性。在一些强调人的因素的用户中，高层可用性测试方案使用的不太普遍，但是如果要了解用户初次访问你的应用程序或它的主要功能时他们的感觉，这些方案是很重要的。

VoiceXML应用程序与Web站点的情况很相像。如果用户对应用程序的印象不好，或者他们发现应用程序不如想像的那样有价值，那么他们就会很快放弃这种应用程序。由于一旦用户放弃使用某种应用之后，要想再次吸引他们来使用该应用程序，这几乎是不可能的，因此你们必须尽一切可能给用户造成一个良好的初步印象。

高层可用性测试方案应该搞得简单一些。其目标是让用户能够试探性地访问你的应用程序，使你能够观察用户如何作出反应。表6-6列出了对VoiceXML图书和T恤衫销售站点进行高层可用性测试的方案举例。

表6-6 高层可用性测试方案举例

测试内容	说 明
访问应用程序	<p>向用户简单介绍一下测试的要求：“你将拨号访问一个语音应用程序，该应用程序用于向你销售图书和相关的产品。请拨打电话号码，然后收听提示。如果你愿意的话，你可以试一下。你可以随时挂断电话。”</p> <p>这项测试的目的是了解用户对应用程序的“心理反应”，观察他们对应用程序的运行情况说些什么、做些什么，会很快地放弃使用该应用程序，还是浏览该应用程序的整个运行过程</p> <p>测试之后立即访问测试人员，了解他们对应用程序的功能、使用是否方便、声音质量等问题的看法</p> <p>将每个可用性问题记录下来，供以后分析之用</p>
浏览主菜单	<p>向用户简单介绍一下测试的要求：“你将拨号访问该语音应用程序，并且通过VoiceXML应用程序购买一本图书。如果你有什么问题，你可以设法继续进行操作，也可以挂断电话。”</p> <p>这项测试的目的是了解用户执行基本浏览操作的能力，观察他们对应用程序的运行情况说些什么、做些什么，他们是选用口语输入还是使用DTMF？他们看上去是否搞不清楚？他们立即挂断电话了吗？</p> <p>测试完成后立即询问测试人员，了解他们对购买图书的过程有些什么想法</p> <p>将每个可用性问题记录下来，供以后分析之用</p>
选项测试	<p>你可能希望能够接着进行一些高层选项测试。例如，如果有所有测试人员都使用DTMF，那么这是“购买图书”应用程序测试的一个替代版本，它规定使用口语输入。如果所有测试人员都使用口语输入，那么这是“购买图书”应用程序的一个替代版本，它规定使用DTMF输入</p>

2. 低层可用性测试方案

低层详细的可用性测试方案用于测试应用程序中的特定活动，一般来说，设计方案的测试可以确保应用程序中的每项活动均得到测试。

除了测试应用程序的基本活动外，应用程序其他的功能也应该专门进行测试，这些功能包括：

- 访问帮助信息。
- 应用程序中使用的所有声音（TTS和录音）的可懂度和必要性。

- 出错消息和错误的恢复。
- 应用程序等待数据处理或外部数据检索期间，播放补空信息（比如“填空”声音或广告）的可接受的程度和实用性。
- 关键任务通过应用程序的通常次数的可接受程度。

这些功能可以在你为应用程序的基本活动设计的测试方案中进行测试，关键问题是它们被测试。

低层可用性测试方案举例

每个测试方案我们通常要建立两个版本：

- 一个简单的用户版本，使用用户术语来说明任务。
- 一个比较详细的版本，供可用性测试人员使用。它包括应用程序的安装、你想要明确计量的时间、预先准备询问的问题，以及进行记录的间隔时间等特定要求。

测试方案的用户版本包括下列操作步骤：

- 1) 通过电话号码1-800-555-1212访问应用程序。
- 2) 使用该应用程序，以便将一本图书和一件T恤衫添加给你的订单。
- 3) 从你的订单中删除T恤衫，并且完成图书的购买。

测试方案的测试人员版本包含下列操作步骤：

- 1) 检查以确保该应用程序已经启动并且运行正常。
- 2) 确保用户既选定了图书，也选定了T恤衫。
- 3) 确保用户在完成购书前删除了T恤衫。
- 4) 请助手对从成功地连接到应用程序到完成购物的整个过程进行计时。
- 5) 观察从订单中删除商品时出现的任何问题。
- 6) 询问下列问题：

“你是否清楚从订单中删除T恤衫时你需要进行什么操作？”

“当你删除T恤衫后你是否搞不清楚如何完成图书的购买？”

6.5.4 使用测试结果来改进应用程序的设计

你必须调查5至8个人，才能确定你已经发现了应用程序中很高百分比的可用性问题。

你的可用性测试结果构成了测试过程中观察到的可用性问题。例如：

- “用户不理解主菜单的含义是什么，因此挂断了电话。”
- “用户不小心访问了DTMF，但是不理解出错消息”。
- “用户说，当他听到帮助消息时，觉得莫名其妙。”

如果你拥有一个典型的应用程序，并且有一到两个人帮助你进行测试情况的观察，那么最后你可能遇到上百个或者更多的可用性问题。你必须采用一种很好的方法，将各种结果综合起来，与应用程序开发小组的其他成员一道确定如何使用测试结果来改进你的应用程序。

下面简要地概述了对可用性问题进行分组，给问题确定优先级，并且在解决方案上取得一致意见的过程：

- 使用相似性分析法，将关系密切的各种问题分门别类组合在一起。

- 去掉重复的问题。
- 每个组的成员都应该给每个可用性问题赋予一个优先级号码，号码1表示该问题必须在用户接触该应用程序之前解决，2表示这是个很重要的可用性问题，应该在应用程序投入使用前解决，3表示该问题可以在推出应用程序的新版本时再加以解决，4表示这不是个可用性问题。
- 如有必要，可以通过讨论来解决在问题的优先级上存在的不同意见。
- 制定相应的原则，以便解决优先级为1的所有问题和可以在可用时间内解决的优先级为2的所有问题。

6.5.5 完成可用性测试的工作周期

一般来说，应用程序的可用性测试和以用户为中心的应用程序的设计是个重复操作的过程。在应用程序投入实际部署之前，对VoiceXML代码原型进行的可用性测试方法至少应该重复进行一次，以确保在以前发现的可用性问题得到解决时，应用程序中不会冒出新的可用性问题。

结束以用户为中心的应用程序开发过程

与可用性测试相关的以用户为中心的应用程序设计周期的最后一个操作步骤是对用户进行跟踪。

如果你按照这个工作过程进行应用程序的开发，你应该能够得到一个非常出色的应用程序。最后你要对后续情况实施跟踪，以了解你的用户是否同意部署该应用程序，并且在这个过程中获取有关的输入数据，帮助你把应用程序设计得更好，为应用程序设置新的特性，并且确定潜在的新的应用程序。

对用户进行跟踪的方法很多，比如：

- 采访。
- 调查。
- 使用你的应用程序或平台中内置的自动化数据收集工具。

不管你使用何种方法，重要的问题在于要了解用户喜欢应用程序中的哪些功能，不喜欢哪些功能，它们希望将来增加哪些功能。

6.6 小结

本章介绍了用户分析和任务分析，这是与开发人员使用的与应用程序的开发相关的两项重要工作。

用户分析用于取得应用程序的目标用户群的特征，使应用程序的设计能够明确地满足他们的需求，并且能够吸引他们使用应用程序。

任务分析用于确定应用程序应该实现自动化或者使之能够执行的工作。出色的任务分析能够用来定义应用程序的范围和功能。

本章讲述了如何开发高级和中级应用程序的原型，并且说明了如何评估和测试应用程序原型的可用性。本章最后介绍了对VoiceXML代码原型进行全面的可用性测试的过程。

第7章 VoiceXML应用程序的开发过程

开发VoiceXML应用程序必须采用符合严格规则的方法，这与传统的迭代开发方法不太相同。VoiceXML应用程序的开发和部署可以分为下面几个主要功能领域：

- 要求
- 结构
- 设计和实现
- 性能计量参数的定义
- 测试
- 部署

这些功能领域在已有的应用程序开发环境中是通用的。但是由于应用程序开发时使用的实现语言不同，因此它的实现方法也各有差异。本章并不试图定义你已经知道的各种术语，相反，当软件项目的每个功能领域适用于VoiceXML应用程序的开发时，我们将要充分介绍这些功能领域。

7.1 确定应用程序开发的要求

在确定应用程序开发要求的阶段中出现的部分问题已经在涉及 VUI设计的各章中作了重点介绍。正如这些章节中所说的那样，VUI的设计决定了最终用户对VoiceXML应用程序将会有什么样的感觉。显然，最终用户的感觉将决定你的应用程序设计的成败，不过这要看VoiceXML应用程序是否在国际上加以部署应用。为一种口语开发的VUI显然无法完全重复用于另一种口语。文语转换（TTS）语言使用的情况如何呢？你是否知道你打算部署应用程序的国家的各个地区说的是何种语言（或方言）呢？

对于采用VoiceXML的各种应用程序的开发，你必须解决许许多多的问题。下面各节将要综合介绍在你实际开发应用程序之前就必须解决的一些关键问题。

7.1.1 应用程序部署地区使用的口语

由于语音识别程序通常必须根据特定语言的情况进行微调，因此口语的地区性差异将会影响这些语音识别程序的识别率。而同样的用户对设计上相似的TTS软件工具提供的TTS合成语音的理解是不一样的。

如果使用口语作为应用程序的营销条件之一，那么应用程序的设计人员应该考虑使用大多数人所说的语言。在任何情况下，应用程序的设计人员都应该在应用程序的设计取得较大进展之前，着手对应用程序进行初步的测试，以便确定语音识别率和合成语音的可懂度。应用程序设计人员应该确定应用程序部署的大多数地区和国家所说的语言。然后，设计人员应该寻找在自动语音识别（ASR）和TTS软件工具上支持的语言的语音门户提供商或托管商。如果需要多种

语言的界面，那么选用的语言不仅仍然应该反映应用程序部署地区的语言使用情况，而且界面的设计也必须考虑说多种语言的人使用上的需要。

7.1.2 语法的定义

为应用程序定义的语法在用户的对话技术中起着非常重要的作用。语法的定义与提示选项一道，构成了语音用户界面的基础。

语法的定义应该做到语法的匹配点尽可能明确。互相非常接近的语法匹配点，比如“Dawn”和“Don”，可能映射为同一个词组，从而会降低语音识别率。如果支持多个有效的语法，将会增加语音识别上的模糊度。当多个语法同时有效时，各个不同语法之间的重叠应该尽可能减少。之所以必须做到这一点，是因为大多数语音识别程序无法确定语法词组出现在哪个上下文环境中，将返回词组列表中的第一个匹配词组。

7.1.3 语音识别与TTS软件工具的选择

上面讲过，最终用户使用的口语对自动语音识别（ASR）和TTS软件工具运行的有效性有着很大的影响。因此必须非常注意语音门户提供商使用的软件工具。在最理想的情况下，ASR和TTS软件工具是根据相同的语言模式开发的。但是，由于同一个制造商同时开发两种类型的软件工具的情况非常少，因此各个软件工具之间必然存在某些差异。

你应该设法彻底弄懂语音门户使用的语音处理软件工具的文档。这包括语音应用程序供应商和语音门户提供商支持的语言和使用的测试/检验策略，以及用于ASR软件工具的环境噪声模型的类型。这最后一个问题很重要，因为有些ASR软件工具供应商是根据录音工作室的话筒录音来使用环境噪声模型的。理想的情况是，有线电话必须使用根据有线电话建立的电话环境噪声模型，而无线电话则必须使用根据无线（移动）电话建立的环境噪声模型。每种环境噪声模型都是根据每种类型的电话网的特性来建立的。

7.1.4 支持的VoiceXML应用程序技术规范

开发VoiceXML应用程序时，VoiceXML应用程序的技术规范达到的水平，以及应用程序符合VoiceXML技术规范的程度，都是必须考虑的重要问题，因为它支持的特性决定了开发的应用程序能够达到的技术水平的上限。更为重要的是，某个供应商究竟如何将技术要求转换成特定平台的VoiceXML元素。VoiceXML论坛的规范一致性小组将在本书撰写时期内发布一个用于VoiceXML2.0技术规范的一致性测试软件。就技术规范本身的演变发展情况而言，截止到撰写本书时，VoiceXML2.0技术规范已经从工作草案变成了建议实施的标准。

请对你感兴趣的若干个供应商的VoiceXML技术规范的演变发展路子进行对比。对于特定平台的VoiceXML元素，请在你可以使用的任何硬件平台上对这些元素进行测试，以便了解各种实现方法之间的差异。在某些情况下，你必须在Nuance、IBM、VoiceGenie或tellme等门户网站进行了注册。

7.1.5 进行数据上的准备

进行数据上的准备就是将完成电话呼叫所需要的数据装入语音门户的过程。这些数据包括

用户身份标识号、原始呼叫方的电话号码、被呼叫方的电话号码，以及电话被转接到的号码。数据准备涉及到电话号码的时候，应用程序设计人员至少应该对电话号码的编号方案有一个大致的了解。电话号码的编号方案是指网络服务提供商（CLEC、长途电话运营商等）的交换机能够接续的电话的号码有效范围。凡是在这个范围之外的号码都不允许完成电话的呼叫，这通常会产生一个电话忙音。

语音门户提供商应该提供允许进入门户的拨号电话号码的详细情况，还要提供拨号电话号码范围的详细情况。有些提供商甚至使用全球的8XX电话，而每种服务都有它自己专用的访问代码。如果是离开门户的转接电话，应用程序设计人员应该确保在试图打电话之前就存在用于对电话进行筛选的逻辑方法。如果电话用户通过语音界面或者Web界面提供了转接电话号码，那么该界面应该负责进行电话的筛选工作。如果是一般的基于电话用户的服务，应用程序设计人员必须定义送往语音门户提供商的数据格式和预期的数据流。然后，语音门户提供商就可以利用它与网络提供商的身份，出色地满足VoiceXML应用程序的需要。

7.1.6 内容提供商界面

VoiceXML应用程序最重要的特性之一是它具备访问内容的功能。如何将信息显示给接收端的最终用户，这个问题已经在介绍VoiceXML用户界面的几章中进行了讲解。与内容提供商之间的界面对最终用户使用应用程序的操作技巧的这些感觉都会产生很大的影响。

因特网上的数据通常存放在数据库中，也可以根据某些操作的编译数据在移动中派生出来。有些内容提供商允许通过Java数据库连接（JDBC）访问他们的数据库，不过这种访问需要一个在内容提供商的网络上运行的辅助服务器。另外，还有一些提供商拥有一个访问其数据的XML界面。这种界面维护起来容易得多，因为查询和应答只需要符合公认的数据类型定义（DTD）即可，该定义可以作为内容提供商与要求访问的用户之间的界面协议。

与内容提供商协商一个界面协议需要很长时间才能完成。如果语音门户主机与你的应用程序要求的那种内容提供商之间已经有一个很好的界面协议，那么这些门户主机将是最好的选择。另外，如果主机拥有一个可以访问的文件夹，里面包含值得信赖的内容提供商，那么这也增加应用程序在市场上取得成功的机会。

7.1.7 性能、容量和可靠性

性能、容量和可靠性是支持高质量语音门户平台所需要的主要条件。这些特殊要求的实现显然不属于应用程序设计人员的职责范围。但是应用程序设计人员可以使用这些数据来帮助确定能满足其需要的最佳配置平台，下面这个列表包含了VoiceXML应用程序设计人员在设计语音门户的性能时应该考虑的最重要的参数：

- 当基准测试的电话呼叫模型中的TTS，ASR，语音提示和DTMF输入的各种不同组合都有效时，每秒钟的最大电话呼叫次数。请注意，为TTS或ASR选择什么样的语言，将会影响语音门户的容量。由于美国英语通常是基准测试语言，因此基准测试的结果最有可能用该语言来显示。
- 可下载文件的最大长度，它应该包括语音提示文件，VoiceXML页或语法文件。文件越长，

下载时间就越长。虽然使用缓存可以大大减轻文件长度带来的影响，但是门户处理文件的能力显然是有一定限度的。

- 电话用户的最大数量将会影响应用程序安装后能够访问应用程序的呼叫者的数量。预计的电话用户数量以及他们的呼叫模式不仅会影响每秒钟呼叫的能力，而且也会导致提供商增加更多的门户。同样，这些方面的考虑对应用程序设计人员来说是透明的，不过，这些特殊问题会因为某些明显的原因而增加应用程序部署的成本。
- 提示打断等待时间是指中断提示与实际停止提示之间所经过的时间量。由于语音识别不需要识别DTMF音调输入，因此DTMF打断输入的等待时间应该是最小的等待时间。加上人的因素的测试结果显示，打断等待时间不应该过多地超过250毫秒。

上面4个性能参数中，提示打断等待时间最重要，因为它会直接影响最终用户对你的应用程序性能的印象。因此，如果负责处理你的预期电话呼叫量和用户数量的提供商需要正确设置语音门户的容量大小，那么首先要解决好提示打断等待时间这个问题。

7.2 确定应用程序的结构

你的应用程序采用何种结构，主要应该取决于你打算将应用程序部署在什么地方，以及你打算让它干什么。运行繁的应用程序，比如根据地点运行的服务程序，显然比较适用于无线设备和无线网络，而其他服务程序，比如股票行情，气象预报或旅游服务程序，可以在任何基础设施上运行。本书前面几章以很大篇幅介绍了你原来部署应用程序的网络结构的细节和应该考虑的问题。本节将要讲述其他方面的一些考虑。

在无线网络上部署的应用程序将会受到各种网络因素的影响，比如语音编码的类型和信号长度等。应用程序还会受到一些非网络因素的影响，比如无线设备周围的环境噪声，以及说话人与无线设备的话筒之间的距离等。语音编码与解码的方法有助于更好地使用分配给无线设备的载波的频谱带宽。由于语音编码只是接近实际说话的声音，因此它们可能丢失部分原始话音数据。根据无线设备与信号塔之间的距离远近，或者信号通路是否受阻，无线设备与信号塔之间的信号强度可能会忽强忽弱。设备离信号塔越近，信号就越强，并且声音的质量通常越好。除了可以选择最能满足其需求的提供商外，无线设备用户是无法控制上面所说的网络因素的。

无线设备用户可以在一定程度上控制其最终用户对VoiceXML应用程序的使用感觉。控制的方法之一是调整他们使用设备时所处的环境。例如，当用户在行进中的车辆中使用无线设备时，车辆引擎和道路产生的噪声将会明显地影响语音识别的正确性。如果又遇到其他的一些因素，比如车窗打开着，那么语音识别的正确性将会进一步降低。无线设备的话筒位置可能会增加或者减少这些因素产生的影响。如果话筒放在汽车里，就会增加有害因素的影响，而话筒放在手机里边，则会减轻有害因素的影响。比较可行的做法是，用户可以关掉汽车引擎，关上车窗，然后再调用应用程序，不过这是另一个问题了。

前面讲过，VoiceXML应用程序设计人员应该确保语音门户提供商使用的语音识别模型已经考虑到由于无线设备的结构而产生的有害因素。门户本身提供了一种调节语音引擎识别参数的手段，从而可以提高某些应用程序的语音识别率。平台参数能量底线（energy-floor）和最小发音持续时间（minimum-utterance-duration）分别用于定义启动分析之前检测的最小能量和可以

视为有效发音的最小语音持续时间。即使是平台的调节，对应用程序的任何后续调试都应该在无线网络和有线网络中进行。

7.3 应用程序设计的实现方法

到现在为止，我们已经介绍了初步的应用程序开发阶段，这将有助于你为你开发的VoiceXML应用程序选择实际的语音门户提供商。最重要的是你可以确定预期的客户群体说的语言、内容提供商、连接到内容提供商的数据的界面（如果有的话）、服务应用程序预期的用户数量、用于访问服务程序的电话设备的类型等。当这些要求确定以后，现在你就可以转入VoiceXML应用程序的设计和实现工作阶段。

如上所述，第5章介绍了VoiceXML应用程序设计的最关键组件，即语音用户界面（VUI）的基础知识。鉴于VUI具备的重要性，这在相当程度上说明你在应用程序的设计上花费的时间应该多于实现应用程序所需要的时间。无论你的应用程序拥有多少出色的组件，如果语音用户界面设计得很糟糕，那么应用程序必然要失败。应用程序应该拥有某种程度的启发式分析功能，这样，用户就能迅速获得他们需要的信息。例如，最好应该让应用程序始终都能跟踪各种响应，从而使新用户能够得到更多的信息提示，而富有经验的用户则得到较少的提示。为了解决可能出现的噪声环境影响语音识别的问题，应用程序设计人员可以用DTMF输入来增强语音输入。VUI构成了VoiceXML应用程序的前端部分，而内容提供商界面即CPI则提供了应用程序“后端”的基础。

了解数据类型以及如何访问数据，这在很大程度上决定了应用程序需要的界面技术。假设应用程序需要检索本地加油站的汽油价格，信息是通过TTS消息向用户显示的。输入的搜索关键字很可能是搜索半径和汽油等级。返回的信息可能是汽油等级，该等级的汽油每加仑的价格，以及加油站的一个列表。返回的信息将通过TTS软件工具作进一步处理，然后显示给用户。这里，由公认的文档类型定义限制的XML消息将用于查询和应答。现在我们假设该应用程序必须支持向用户显示语音数据和文本数据。

用于无线客户能够使用具备Web访问功能（或不具备Web访问功能）的电话或其他无线设备，如PDA，因此有时需要显示文本数据。XML表单可以用来定义XML页，而XML页既可以转换成VoiceXML页，也可以转换成HTML页。最后，内容提供商显然是用Web页的形式提供“默认的”数据显示方式。因此HTML可能是最后的数据显示类型。

7.4 应用程序性能计量参数的定义

应用程序性能计量参数是个功能强大的工具，它提供的数据不仅可以用来改进用户界面，而且可以用于判断语音门户的质量和可靠性。以用户为中心的应用程序性能计量功能基本上是受应用程序设计人员控制的，并且应该包装成一个重复使用的组件，以便在可能的情况下进一步缩短应用程序的开发周期。下面列出了若干个建议使用的性能计量参数，以帮助用户具体了解应用程序的性能。

- 语音识别率——语音识别率用于表示语音识别程序返回的成功与不成功的语音识别结果的数量。它的作用之一是具体计算成功的语音识别匹配结果的数量和返回的“nomatch”（不

匹配)事件的数量。显然,它的语法定义很重要。“noinput”(无输入)事件可以是个辅助计量值,假定通过用户输入的“超时”属性提供了足够时间的话,可用来测定用户对应用程序说的话的理解程度如何。否则,表示用户在理解提示的含义时遇到了困难。

- 提示与作出响应之间的时间——用户对提示作出响应所花费的时间量是另一个应用程序性能计量参数,它有助于测定提示指令的复杂性。测定提示结束到用户作出响应之间的时间,可以帮助你确定提示的复杂性。如果提示能够帮助用户完成他们的操作的,就表示这些提示是用户感兴趣的,或者表示这些用户是新用户,也可能表示提示包含的信息无法被重复使用提示的用户很容易地记住。不过,由于语音门户在VoiceXML页的解释程序与实际的语音识别程序之间有一个“中间软件”层,因此实际返回时间有一个最多为几百毫秒的容错安全量。
- 断断率——提示的断断率可以定义某个提示是否太长,或者用户是否熟悉该提示传递的信息。提示的断断率还应该与电话的完成率一道进行分析,以便了解呼叫方是与应用程序进行了充分的交互操作,还是遇到困难而挂断了电话。
- 呼叫的持续时间——呼叫的时间长度显然与提示的数量、提示的长度和向内容提供商检索数据的时间等因素有关。请记住,呼叫者能够容忍的电话时间长度随着提供的信息的可用性或收听信息的愿望而变化。这个时间成分最好应该根据其他计量参数来推算。这些计量参数可以作为本参数分析的组成部分。例如,使用移动电话并且根据地点来运行的应用程序将要求尽快向呼叫者显示有关的数据,因为他们很可能以相当高的速度移动其位置。
- 呼叫完成率——本参数用于测定呼叫者输入应用程序后进行呼叫的完成率。它的首要目标是确定在应用程序中的哪些位置上呼叫者能够获得他进行呼叫会话所需要的信息,从而“确保”呼叫者执行切断连接的操作。第二个目标是计算所有其他呼叫者执行切断连接的操作的数量。从应用程序的角度出发,这些切断连接的操作是“不必要的”。
- 提示的数量——提示的数量通常用于提供一个呼叫会话期间呼叫者遍历的提示的总数。当人们对一个典型的或理想的呼叫进行基准测试,然后将这个结果与实际呼叫的测试值进行比较时,本参数的作用很大。
- 请求提示至播放提示之间的延迟——本参数可以用于确定语音合成和语音播放资源的有效性。虽然可建立“快速”缓存,但是将输入数据传递给语音播放资源仍然需要花费一定的时间。因此,最终用户将会受到各种可以预见的延迟的影响。

上述每个参数如果作为孤立的数据来使用,那么不会产生足够的信息。相反,如果将这些参数组成不同的组合,你就会对一般用户对应用程序的感觉有一个清楚的概念,例如,假设用户的一系列电话呼叫出现了异乎寻常的大量打断现象,并且许多呼叫者挂断了电话。现在我们假设呼叫的平均持续时间小于对应用程序呼叫的正常持续时间。将这3个计量参数组合起来,可能意味着由于联系不畅,从而导致呼叫者不到时候就切断了与服务的连接。一般来说,你可以通过认真分析这些计量参数,对应用程序进行相应的调整,从而大大提高界面的可用性。平台的性能计量参数,如呼叫的尝试率和呼叫的失败率、VoiceXML页的下载率、声音文件、语法文件,都可以作为以用户为中心的应用程序性能计量参数的补充。不过,由于这些都是平台的性能计量参数,因此应用程序时设计人员应该向门户提供商索取这些参数。

性能计量参数既可以是在语音门户上测定的参数，也可以是在VoiceXML应用程序上测定的参数，前者可能必须与门户提供商协商之后才能进行数据的收集，而性能计量参数本身是应用程序设计人员无法控制的。此外，对数据进行配置和索取可能要投入一定的费用。对应用程序的性能计量参数进行收集，这是人们首选的方法，因为这可以使应用程序设计人员能够更好地控制性能参数。

7.5 VoiceXML应用程序的测试

从概念上讲，对VoiceXML应用程序进行测试，与对其他类型的软件进行测试是相似的，但是对VoiceXML应用程序进行测试要更复杂一些。因为你需要从语音用户的角度来测试应用程序的可用性。你的测试方案应该建立若干项前提条件，并且必须得到语音门户提供商的认可。前提条件之一是语音软件工具必须在宣称采用的各种语言领域中进行彻底的主体测试。所谓主体测试，就是由实际的测试主体（测试人）来检验应用程序的有效性。在这里，就是要测试语音识别和TTS软件工具的有效性。在理想情况下，测试主体（测试人）使用的语言应该是一种具有代表性的典型语言（包括人们普遍使用的方言），并且能够流利地说这种语言。这些主体测试应该已经由语音设备供应商进行过了，并且又被语音门户供应商进行过，这些测试都是他们进行系统集成工作的组成部分。如果这些前提条件不具备检验作用，那么你应该与语音门户提供商合作进行应用程序的可用性测试，或者认真考虑选择另一个供应商来进行测试。

前面介绍的正规测试方法可以用于测试VoiceXML应用程序，不过对应用程序进行测试的最容易（和最便宜）的方法是将它投入实际应用。在应用程序的性能计量参数收集功能启动之后，正常使用应用程序就能够提供关于应用程序的很好的测试信息。因此对这些参数数据的分析可以用做改进应用程序的反馈循环。当应用程序达到一定的成熟程度时，就可以关闭参数收集功能，因为在收集过程中会影响性能。不过，在应用程序投入实际应用之前，应该进行一定程度的单元组件测试。在整个系统的后端，内容提供商的界面必须进行测试，但是它可以抽出来进行初始可用性测试。

内容提供商界面应该根据通过界面的数据值的预计范围进行最少的单元测试。用于实际界面的技术，无论是RMI、CORBA、SOAP，还是XML，都会增加测试的复杂性。由于VoiceXML应用程序是一种时间敏感型应用程序，它的主要目标应该确保尽快向用户显示数据。因此端到端的性能测试就变得十分重要。

进行全面的主体测试对于大多数VoiceXML应用程序的开发来说成本太高，很难进行，而端到端的性能测试也是如此。进行应用程序测试的最好办法是使用用户一端的发出呼叫的电话驱动程序，该驱动程序能够支持语音输出，以便驱动语音识别程序。更加实用的方法是抽出语音界面，再使用该用户界面的软件驱动程序。应该注意的是，如何层次上的性能测试都必须在语音门户提供商与内容提供商之间实现时间同步和进行相应的协商。这些安排最终应该根据应用程序部署后预计产生的经营收入来加以决定。

7.6 应用程序的部署

你已经根据客户需求，进行了应用程序的设计，又对程序进行了运行实现，并且进行了测

试。现在你必须将它提供给广大用户使用。前面我们介绍了某些门户提供商如何提供带有8xx的电话号码的账户，这些电话号码可以用于访问VoiceXML应用程序。如果你的应用程序是以用户为基础来运行的，那么这时你必须着手对用户进行数据准备。最好采用自动数据准备操作过程，不过为应用程序定义的这个过程必须符合语音门户提供商和网络提供商的要求。如果应用程序设计人员打算用他的应用程序来谋取经营收入，那么他必须进一步考虑应用程序的技术支持和维护方面的问题。

7.7 小结

本章讲述了对VoiceXML应用程序进行设计、实现、测试和部署等方面的详细过程。我们采用的方法是假设应用程序的部署将带有为应用程序设计人员带来经营收入的意图。本章介绍的内容决不是包罗万象的，不过至少可以为你提供足够的信息，以便确定建立能够获得经营收益的VoiceXML应用程序的可行性。

第8章 VoiceXML的基本命令

本章将要介绍建立应用程序文档、定义变量和在文档间导航时使用的VoiceXML元素的定义。本章还要讲述每个VoiceXML元素的属性和它们的值。我们首先要用本章介绍的VoiceXML元素建立一个语音商务站点。通过该站点，用户可以订购图书和T恤衫，然后根据购买的商品的质量和用户是否拥有电子购物优惠券来计算订单上商品的费用。

8.1 第一个VoiceXML程序

首先让我们对VoiceXML语言进行一次全面的考察，就像对编程语言的所有考察都需要一个“Hello World”（问候程序）一样。我们要对该程序的输出稍作修改，使它成为“Welcome to Sams Publishing's online bookstore”（欢迎你访问萨姆出版社的在线书店），而不是简单的“Hello World”，这样就更适合该应用程序的需要。这个Welcome.vxml文件包含的程序类似下面的样子：

```
<?xml version="1.0"?>
<!--Welcome.vxml -->
<!--Hello World -->
<vxm1 version="1.0">
  <form id="welcome">
    <block>
      Welcome to Sams Publishing's online bookstore
    </block>
  </form>
</vxm1>
```

当该程序页被执行时，它只需要使用文本到语音（TTS）的转换工具来说出“Welcome to Sams Publishing's online bookstore”。下面让我们更加详细地说明该程序的代码。

8.2 xml元素

代码行`<? xml version = “1.0” ?>`向XML解析器或编辑器指明该文档是用XML 1.0语法编写的。所有VoiceXML 1.0文档都是用XML 1.0版语法编写的。虽然元素`xml`对于VoiceXML解释程序来说并不是必不可少的元素，但是最好应该使用该元素，这样就可以检查VoiceXML文档的格式是否正确，是否符合VoiceXML文档类型定义（DTD）。

8.3 注释

VoiceXML程序的注释使用标准的XML语法来编写。所有VoiceXML注释一律采用`<!--comment text -->`这个格式。

8.4 vxml元素

所有VoiceXML命令都封装在<vxml>...</vxml>元素中。

属性version负责告诉解释程序，在执行程序文件时，使用哪个VoiceXML文档类型定义(DTD)。本书的所有程序举例和语法都引用网址<http://www.voice.xml.org/voicexml1-0.dtd>上找到的VoiceXML 1.0文档类型定义。

属性application是VoiceXML文档的根文档的URL。当你上载一个文档时，它的根文档也同时进行上载。如果像本例中那样不包含属性application，那么VoiceXML文档就没有根文档。根文档及其用途将在本章中的8.8.9“根文档”这一节中进行更加详细的介绍。

8.5 form元素

我们在第1章中作过这样的定义，VoiceXML对话框用于描述VoiceXML应用程序对用户所说的各种提示，定义和收集用户作出的响应，并且描述程序控制的流程。对话框分为两种，一种是菜单，另一种是窗体。第9章介绍的是菜单对话框。form元素负责执行对话框定义中描述的所有操作。它用于封装与用户的输入和输出相关的命令，在文档中导航并且可以导航到其他的文档，也可以声明变量的值并且将值赋予变量，还可以进行录音，并将电话转接到另一方。

属性id用于标识窗体。在上面这个例子中，我们给窗体赋予的id值是welcome。赋予id的值将在goto、submit和subdialog元素从另一个对话框导航到该窗体时使用。

属性scope用于指明窗体中声明的任何语法的范围。语法将在第9章中详细介绍。

8.6 窗体中的项目

field、record、transfer、object、subdialog、block和initial等元素都称为窗体项目。所有窗体项目都有一些共同的特征。首先，所有窗体项目都封装在form元素之间。其次，所有窗体项目都拥有下列属性。

8.6.1 name属性

name是指与窗体项目元素相关的一个变量，它用于指该变量的值。

8.6.2 expr属性

expr是name属性中声明的变量的初始值。expr既可以是用于求得某个值的表达式，也可以是个实际的值。expr属性并不是必需的。如果省略了expr属性，那么变量将初始化为undefined(未定义)值。

8.6.3 cond属性

cond属性是个用于求得true或false值的表达式。它是个布尔值，用于确定封装在窗体项中的指令将在访问窗体项时被执行。cond的默认值是true。

8.6.4 使用说明

你可以对name、expr和cond值进行相应的操作，以便确定封装在窗体项元素之间的指令是否在访问窗体项时被执行。当多个 field元素封装在一个窗体中时，这个方法最常使用。在这种情况下，每个field的name、expr和cond值都可被操作，以便根据用户对前面问题的答案，确定对用户应该提出什么问题。在第9章中可以找到这种情况的一个例子。循环程序也可以用VoiceXML来编写，方法是对block元素中的name和expr值进行相应的操作，用这种方法建立的循环举例可以在本章中的“循环”小节中找到。

如果name和expr这两个值都作了定义，那么用block元素封装的指令要等到与该代码块相关的变量被清除时才予以执行（这个问题将在本章中的“clear元素”这一节中介绍）。例如，下列封装的指令将因为后续元素不能执行；

```
<field name="field_var" expr="true" >
<field name="field_var" expr="false" cond="false">
<field name="field_var" expr="false" cond="true">
```

如果expr属性省略了，那么不管指令是否封装在窗体项元素中，都将根据cond的值来执行。如果cond的值是true，或者被省略，那么窗体中的指令将被执行；如果cond的值是false，那么窗体项中的指令将不执行。下面这些元素的指令将能够执行：

```
<field name="field_var" cond="true">
<field name="field_var">
<field cond="true">
```

下面这些元素的指令将不能执行：

```
<field name="block_var" cond="false">
<field expr="false" cond="true">
<field cond="false">
```

8.6.5 block元素

block元素是个窗体项。它用于封装相应的语句。这些语句可用于对变量的值进行输出、浏览、声明和赋值。在上面的示例程序中，block元素用于输出变量的值。本文“Welcome to Sams Publishing's online bookstore”使用文本语音转换工具被用户读取。

8.6.6 name属性

name是指与block元素相关的变量。name 的值可以被引用代码块的goto 之类的其他元素使用。name属性不是必不可少的。

8.6.7 expr属性

expr是name属性中声明的变量的初始值。expr不是必不可少的属性，如果它被省略，变量的值便是undefined（未定义）。

8.6.8 cond属性

cond属性是个布尔值，用于确定封装在block元素中的指令是否在访问block时予以执行。cond属性的默认值是true。

8.6.9 使用说明

在上面的示例程序中，block元素没有任何属性，block元素封装的指令将在访问block元素时予以执行。

8.6.10 字段项

字段项是窗体项的子集，它包括field、subdialog、object、record 和transfer等元素。filled、property、grammar、dtmt、prompt和 catch等元素都封装在字段项元素之间。字段项也可以拥有与其相关的影子变量。

8.6.11 filled元素

filled元素既可以是字段项的子元素，也可以是form元素的子元素。当filled元素是字段项的子元素时，封装在filled元素中的可执行代码将在用字段项元素的 name 属性声明的字段项变量被赋予一个非undefined值时予以执行。关于变量何时被赋予一个值以及变量的举例，请参见每个字段项元素的元素小节中的说明。

当filled元素是窗体（form）元素的子元素时，封装在filled元素中的可执行代码将在字段项的各个变量的组合被赋予一个非 undefined 值时予以执行。变量的组合将取决于filled元素的各个属性值。用作窗体的子元素的filled元素的实例可以在第9章中找到。

1. namelist属性

namelist属性只有当 filled元素是 form元素的子元素才使用。namelist属性包含了filled 元素等待被赋予一个值时的所有字段变量。它的默认值是父窗体中的所有字段变量。

2. mode属性

mode属性只有当filled 元素是form元素的子元素时才使用。Mode属性可以使用的值有2个，即 all或any 。如果mode属性的值是any，那么当filled元素的namelist属性中的任何变量被赋予一个非undefined值时，被filled元素封装的可执行代码将被执行。如果mode属性的值是all，那么当filled元素的namelist 属性中的所有变量被赋予一个非undefined 值，并且有一个字段变量已经被最后用户的输入所定义，那么filled元素封装的可执行代码将被执行。

8.7 meta元素

VoiceXML中的meta元素与 HTML中的meta 元素相同。它用于设定关于VoiceXML 文档本身的有关信息，而不是关于文档内容的信息。meta元素用于设定一个属性，并且为它赋予一个值。程序清单8-1中的代码将一些meta元素添加给了“Hello World”这个示例程序。

程序清单8-1 Welcome.vxml

```

<?xml version="1.0"?>
<!--Welcome.vxml -->
<!--Hello World -->
<vxml version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
          content="Welcome page for Sams online bookstore to access call
        ↪(xxx)-xxx-xxxx"/>
        <meta name="keywords" content="VoiceXML Sams Books"/>
        <meta name="maintainer" content="johnnyo@sams.net"/>
        <meta name="robots" content="index,follow"/>
        <meta http-equiv="Date"
              content="Thu, 7 Aug 2000 20:20:20 EST"/>
        <form id="welcome">
            <block>
                Welcome to Sams Publishing's online bookstore
            </block>
        </form>
</vxml>

```

8.7.1 name属性

name 用于设定标记中描述的属性的名字。下面这行代码

```
<meta name="author" content="John O'Reilly"/>
```

用于描述一个带有名字author的属性，并且为该属性赋予值John O'Reilly。

name 属性最常用的值是author、copyright、description、keywords、maintainer 和robots。

属性author用于说明VoiceXML文档的作者。属性 copyright用于说明与文档相关的版本信息。属性description和keyword供搜索引擎用来识别文档。属性maintainer包含了一些联系人的信息，以便用户使用文档时遇到疑问或问题时可以找人询问。属性 robot 包含搜索引擎的网上机器人的信息。这些网上机器人通过使用嵌入Web页中的各个链接，从一个Web页转到另一个 Web页，以便查找因特网上的有关 Web页。在代码举例中，content的index、follow值告诉网上机器人，它拥有查找该Web页的许可权，并且可以按照该页上的链接去搜索别的 Web页。

8.7.2 content属性

content包含name属性中描述的属性的值。例如，在我们的welcome文档中，下面这个代码行

```
<meta name="author" content="John O'Reilly" />
```

为author属性赋予了John O'Reilly这个值。

8.7.3 http-equiv属性

http-equiv属性包含了HTTP 应答报头的名字。HTTP应答报头是浏览器发送文件访问请求时

服务器返回的一个结果信息。与 name属性一样，http-equiv可以与 content 属性结合起来使用。http-equiv 可以代替name 来使用，但是不能与name并用。在示例程序中，下面这行代码

```
<meta http-equiv="Date" content="Thu, 7 Aug 2001 20:20:20 EST" />
```

将HTTP应答报头Data与值 “Thu, 7 Aug 2001 20: 20: 20 EST” 关联了起来。

8.8 变量

VoiceXML变量相当于ECMAScript变量，两者之间只存在很小的差别。与ECMAScript不同的是，VoiceXML变量不能以下划线字符（_）开头，因为用这个字符开头的变量是供内部使用的。标准变量有下面6种：

- Undefined（未定义）——变量被赋予一个值之前它的默认值。
- Null（空）——它只能拥有null值。
- Boolean（布尔型）——它只能拥有true 或false值。
- String（字符串）——用单引号封装的字符串。
- Number（数字型）——任何正整数或负整数，或者十进制值。
- Object（对象型）——一组没有排序的属性。每个属性都包含一个名字、一个值和一组属性。

请看程序清单8-2所示的VoiceXML文档，它包含在文件CalculateCost.vxml中。文件CalculateCost.vxml是在线书店应用程序的一个组成部分，用于计算在线图书订单的购书费用。

程序清单8-2 CalculateCost1.vxml

```
<?xml version="1.0"?>
<!--CalculateCost.vxml -->
<!--This document demonstrates variables by calculating the purchase cost of
--a book order -->

<vxml version="1.0" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright"
        content="Sams Publishing 2001"/>
  <meta name="description"
        content="Calculate the cost of a book order, to access
--call (xxx)-xxx-xxxx"/>

  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <var name="price" />
  <var name="order_prompt" />

  <form id="calculate_price">
    <var name="quantity" expr="2" />
    <var name="unit_price" expr="24.95" />
```

```

<var name="order_text"
      expr="'The total price of the order is ' " />
<block name="assignment" >
    <assign name="price" expr="quantity * unit_price" />
    <assign name="order_prompt" expr="order_text + price" />
</block>
</form>
</vxml>

```

8.8.1 声明变量

在VoiceXML中，变量必须在引用之前加以声明。声明变量可以采用两种方法。首先，可以使用下一节中介绍的var元素来声明变量。变量也可以用窗体项比如block元素进行声明。你可以用name属性在窗体项中声明一个变量，比如 `<block name = “assignment” >`。这个指令声明了一个赋值变量，并且将该变量与block元素关联了起来。

在文档CalculateCost.vxml的代码行`<block name = “assignment” >`中，变量assignment是在代码执行过程中到达block元素时建立并且初始化为undefined的。使用窗体项的name属性声明的变量可以初始化为一个使用窗体项的expr属性产生的值。例如，若要在CalculateCost文档的calculate_price 窗体中将block变量 assignment初始化为false 值，请将下面的代码行

`<block name="assignment" >`

改为

`<block name="assignment" expr="false">`

不过请记住，正如本章中的“block元素”这一节所示，这个赋值操作会导致block元素内的指令无法执行。

var元素

8.8.2 name属性

name属性用于描述变量的名字。它用于引用变量的值。

8.8.3 expr属性

expr属性用于对变量的值进行初始化。expr属性即可以是个求取一个值的表达式，也可以是一个实际的值。如果expr属性被省略，变量将被初始化为值undefined。

8.8.4 使用说明

在上面这个示例代码中，可以使用var元素来声明5个变量。price、order_prompt声明为变量，但是它们没有初始化因此它们的值是undefined。quantity、unit_price 和order_text等变量在声明时进行了值的初始化。quantity初始化为数字2，unit_price初始化为数字24.95，而order_text则初始化为字符串 ‘The total price of the order is’。

assign元素

assign元素用于将新值赋予变量。

8.8.5 name属性

name属性是指新值被赋予给的变量的名字。

8.8.6 expr属性

expr属性是变量的新值。expr属性既可以是用来求取某个值的表达式，也可以是个实际的值。

8.8.7 使用说明

在程序清单8-2中，下面这行代码

```
<assign name="price" expr="quantity * unit_price" />
```

首先计算quantity * unit_price，并且将计算得出的数字结果49.9赋予变量 price。下面这个代码语句

```
<assign name="order_prompt" expr="order_text + price" />
```

将字符串order_text与数字price连接起来，然后将产生的字符串‘The total price of the order is 49.9’赋予变量order_prompt。第9章将要向你介绍如何将49.9作为美元金额来输出。当字符串与数字结合在一起时，产生的结果将是一个字符串类型变量。

注意 关于有效的ECMAScript/VoiceXML操作以及对各种类型的变量执行这些操作所产生的结果类型的完整列表，请参见网址<http://www.jsworld.com/ecmascript/>上的ECMAScript教材中的第9章至11章的内容。

8.8.8 关于变量的其他信息

本节将要进一步展开来说明本章到现在为止介绍的变量的基本概念，并且要介绍一些更加高级的概念，比如变量的分类，变量的作用域，通过根文档来声明应用程序的变量、会话变量、影子变量和变量命名冲突等。

1. 变量的分类

VoiceXML中的变量并没有十分严格的分类。一个已经声明的变量将包含它被赋予的expr值，而不管该变量原先属于什么类型。例如，如果一个变量已经被声明，并且被初始化为整数值2，那么以后也可以合法地将字符串值“two”赋予该变量。下面是一个例子：

```
<!--legal form -->
<form id="test_typing">
  <var name="int" expr="2" />
  <block>
    <assign name="int" expr="'two'" />
  </block>
</form>
```

2. 变量的作用域

在VoiceXML中，可以在整个VoiceXML文档和一个文档的根文档中声明各个变量。变量的

声明的位置决定了变量的作用域。除了在根文档中声明的变量外，变量的作用域的所有逻辑都封装在它的父变量的元素中。例如，如果一个变量是在一个block元素中声明的，那么该变量可以被封装在block元素中的所有指令进行引用，但是不能被block外面的任何指令引用。

变量可以拥有下列作用域，即会话、应用程序、文档、对话框和匿名等作用域。

- 会话变量由执行VoiceXML文档的平台负责声明并赋予一个值。在某个平台上运行的任何应用程序都可以访问该平台的会话变量。会话变量将在本章的“会话变量”小节中作进一步的介绍。
- 应用程序作用域是指在整个多文档中存在的变量。应用程序变量是作为根文档的vxml元素的子元素来声明的。应用程序变量在上载根文档时进行初始化，在上载文档时存在。应用程序变量可以在根文档中看到，并且可以在声明根文档为根的任何文档中看到。
- 由vxml元素封装的变量拥有文档作用域。拥有文档作用域的变量在文档上载时进行初始化，并且在文档上载过程中继续存在。这些变量可以在整个文档中引用。
- 在form或menu元素中用vars元素声明的变量拥有对话框作用域。另外，窗体项中通过name属性声明的变量也拥有对话框作用域。拥有对话框作用域的变量将在初次执行form或menu元素时进行初始化，并且在执行控件位于对话框中时继续存在。
- 匿名作用域是指在block、filled和catch等元素中声明的变量。拥有匿名作用域的变量在访问封装声明变量的var元素的父元素时进行初始化。当block、filled或catch封装的指令被执行时，这些变量将继续存在。

在程序清单8-2中，变量price 和order_prompt 用var元素来声明。由于var元素封装在 vxml元素中，因此 price 和order_prompt 变量拥有文档作用域。

price和order_prompt变量是在文档作用域中声明的，因此可以在除了calculate_price之外的窗体中引用它。例如，price变量可以被将运费和处理费添加给价格的窗体来引用。Order_prompt变量可以被使用文语转换工具读取变量值的窗体引用。由于Quantity、unit_price和order_text等变量是使用窗体 calculate_price 中包含的var元素声明的，因此它们拥有对话框作用域。

8.8.9 根文档

假设你不仅想在线书店上销售图书，而且想要销售带“VoiceXML进行语音应用程序开发”字样的T恤衫，因为人们对这种商品的需求无疑是极大的。为了适应T恤衫和图书销售的需要，你的VoiceXML应用程序必须同时跟踪你对T恤衫和图书购买的情况和合计购买费用。这是你的简单的购物车上的情况。

为了跟踪你购物订单的费用情况，你在根文档BookStoreRoat.vxml中声明3个应用程序变量，即 total_order_cost，total_books 和total_shirts。程序清单8-3列出了这些变化。

程序清单8-3 CalculateCost2.vxml

```
<?xml version="1.0"?>
<!--CalculateCost.vxml -->
<!--Demonstrates the use of root documents -->
<vxml version="1.0" application="BookStoreRoot.vxml ">
```

```

<meta name="author" content="John O'Reilly"/>
<meta name="copyright" content="Sams Publishing 2001"/>
<meta name="description"
      content="Calculate the cost of a book order, to access
      ↪call (xxx)-xxx-xxxx"/>

<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date"
      content="Thu, 7 Aug 2000 20:20:20 EST"/>
<var name="price" />
<var name="order_prompt" />

<form id="calculate_price">
    <var name="quantity" expr="2" />
    <var name="unit_price" expr="24.95" />
    <var name="order_text" expr="'The total price of the order is '" />
    <block name="assignment" >
        <assign name="price" expr="quantity * unit_price" />
        <assign name="order_prompt" expr="order_text + price" />
        <assign name="application.total_order_cost"
               expr="application.total_order_cost + price" />
        <assign name="application.total_books" expr="application.total_
            ↪books + quantity" />
    </block>
</form>
</vxm1>

<?xml version="1.0"?>
<!--BookStoreRoot.vxml -->
<!--Root document →
<vxm1 version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
          content="Root document for our book store, to access call
          ↪(xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="total_order_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />
</vxm1>

```

根文档使用的语法与其他VoiceXML文档相同。VoiceXML文档可以使用vxm1元素中的application属性来声明它的根文档。

在我们的示例程序中，CalculateCost.vxml中的代码行：

```
<vxm1 version="1.0" application="BookStoreRoot.vxml">
```

将BookStoreRoot.vxml声明为 CalculateCost.vxml的根文档。由于 BookStoreRoot.vxml文件与CalculateCost.vxml 驻留在同一个目录中，因此BookStoreRoot.vxml的完整URL可以省略。如果BookStoreRoot.vxml文件与CalculateCost.vxml不在同一个目录或Web服务器中，那么它的完整URL必须包含进去，不能省略。虽然在这个例子中引用 CalculateCost.vxml 中的total_order_cost 和 total_books 变量时不一定要使用前缀application，但是它有助于程序设计人员将变量识别为应用程序变量。

8.8.10 会话变量

会话变量包含了解释 VoiceXML文档的平台的特定信息。每个平台可以提供不同的会话变量，但是VoiceXML变量session.telphone.ani、session.telephone.dnis、session.telephone.iidigits、session.telephone.uui等则是标准变量。每个平台的标准变量可以由该平台上运行的任何VoiceXML文档进行访问。会话变量的引用方式与其他VoiceXML变量的引用方式相同，不过会话变量不能被赋予新的值，因为它们是只读变量。如果某个平台不支持会话变量。那么该变量的值将是undefined（未定义）。

Session.telephone.ani变量的值是访问VoiceXML页时使用的电话的号码。session.telephone.dnis变量的值是拨号访问VoiceXML页的个人电话号码。session.telephone.iidigits变量用于提供关于（从付费电话、移动电话等发出的）呼叫的信息。session.telephone.uui变量负责提供ISDN呼叫建立协议中包含的信息。

8.8.11 影子变量

影子变量是与字段项相关的变量。字段项是窗体项的子集，它包括field、subdialog、object、record 和transfer 等元素。影子变量用于返回字段项执行的结果。例如，record 元素可以返回影子变量duration，代表录音的时间长度。

影子变量引用时使用的句法是 field_item_name \$.shadow_variable_name。Field_item-name 是字段项的名字，shadow_variable_name是被引用的影子变量的名字。访问名字为 recording 的record元素中的影子变量duration 的句法类似下面的样子：

```
recording $.duration
```

8.8.12 变量命名冲突

不同作用域的若干个变量有可能被声明为相同的名字。若要引用某个特定的变量，可以使用句法<scope>、<variable name>来引用它，这个句法中的scope 是想要引用的变量的作用域。当变量不带作用域这个前缀，并且存在命名冲突时，便使用仍然可以看到的带有限定性最明确的作用域的变量。从限定性最明确的作用域到限定性最不明确的作用域，作用域的顺序依次为anonymous, dialog, document, application和 session。程序清单8-4，显示了一个带有命名冲突

的VoiceXML文档。

程序清单8-4 FunWithHiding.vxml

```

<?xml version="1.0"?>
<!--FunWithHiding.vxml -->
<!--Demonstrates variable hiding -->
<vxm version="1.0" application="FunWithHidingRoot.vxml" >
    <var name="hide" expr="2" />
    <var name="alsohide" expr="5" />
    <form>
        <var name="hide" expr="3" />
        <var name="alsohide" expr="6" />
        <block>
            <var name="hide" expr="4" />
            <!-- hide is 4 -->
            <var name="hide" expr="4.5" />
            <!-- Since hide is already declared in this scope, the second
                var instruction acts as an assignment command. Hide is
                →4.5 →
            <!-- dialog.hide is 3 →
            <!-- document.hide is 2 →
            <!-- application.hide is 1 -->
            <!-- alsohide is 6 here even though document.alsohide and
                dialog.also hide are visible at this point. alsohide
                is 6 because the scope of dialog.alsohide is more
                specific than document.alsohide -->
        </block>
    </form>
</vxm>

FunWithHidingRoot.vxml
<?xml version="1.0"?>
<!--Root document -->
<vxm version="1.0" >
    <var name="hide" expr="1" />
</vxm>

```

8.9 if元素

if元素可用于条件逻辑。如果cond属性的值计算结果为true，那么if元素封装的指令将被执行。如果cond属性的值为false，那么用if元素封装的指令将不予执行。

程序举例

在本章中所用的在线书店应用程序例子中，图书的价格总是24.95美元。假设你想为你的VIP成员提供电子优惠条件，以鼓励他们购买在线图书，那么凡是拥有电子优惠条件的人都可以享受20%的折扣。而其他人则必须用正常价格24.95美元来购买图书。这种情况下的VoiceXML文档将类似程序清单8-5所示的样子。

程序清单8-5 CalculateCost3.vxml

```

<?xml version="1.0"?>
<!--CalculateCost.vxml -->
<!--Demonstrates the if element-->
<vxm version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>

    <meta name="description"
        content="Calculate the cost of a book order, to access call
        &(xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="price" />
    <var name="order_prompt" />
    <var name="isCoupon" expr="true" />
    <form id="calculate_price">
        <var name="quantity" expr="2" />
        <var name="unit_price" expr="24.95" />
        <var name="order_text" expr="The total price of the order is '' />
        <block name="assignment" >
            <if cond="isCoupon == true">
                <assign name="unit_price" expr="unit_price * 0.8" />
            </if>
            <assign name="price" expr="quantity * unit_price" />
            <assign name="order_prompt" expr="order_text + price" />
            <assign name="application.total_order_cost"
                expr="application.total_order_cost + price" />
            <assign name="application.total_books"
                expr="application.total_books + quantity" />
        </block>
    </form>
</vxm>

```

在这个例子中，if语句封装在一个block语句中。if元素即可以封装在 block标记中，也可以封装在catch标记中， isCoupon变量的值被设置为 true，因此下面的语句

```
<assign name="unit_price" expr="unit_price * 0.8" />
```

被执行。变量 unit_price 被设置为24.95美元的80%，即19.96。如果isCoupon 设置为false ，那么下面这个语句

```
<assign name="unit_price" expr="unit_price * 0.8" />
```

将不执行，并且在计算价格时unit_price将保持为24.95。

8.10 if / elseif / else元素

elseif和else元素与if元素一道用作条件逻辑。elseif 与else元素 总是封装在一个if元素之中。

if元素可以包含多个elseif元素，但是只能包含一个 else元素。如果在if元素中发现一个cond属性的值计算为false，那么将计算下一个elseif cond的表达式。这个过程将一直继续下去，直到有一个 elseif元素的cond属性值计算为true，或者所有elseif元素均被访问为止。如果没有一个cond 属性的值被计算为true，则执行else 元素后面的指令。

8.10.1 cond属性

cond是个表达式，它的值既可以计算为true，也可以计算为 false。如果cond 属性的值计算为true，那么封装在if或elseif元素中的指令将被执行。如果 cond 属性的值计算为false，便访问下一个elseif或 else元素。

8.10.2 程序举例

假设你不是提供电子优惠条件，而是为批量商品订单提供各种折扣。如果订购的图书超过10册，那么可以享受20%的折扣。如果订购的图书超过100册。则可享受40%的折扣。对于所有其他的订单，你将按全价24.95美元收费。计算批量折扣的图书价格的VoiceXML应用程序文档将类似程序清单8-6所示的样子。

程序清单8-6 CalculateCost4.vxml

```
<?xml version="1.0"?>
<!--CalculateCost.vxml -->
<!--Demonstrates the if/elseif/else elements-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Calculate the cost of a book order, to access
        <call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="price" />
    <var name="order_prompt" />

    <form id="calculate_price">
        <var name="quantity" expr="2" />
        <var name="unit_price" expr="24.95" />
        <var name="order_text" expr="'The total price of the order is '" />
        <block name="assignment" >
            <if cond="quantity &gt; 100">
                <assign name="unit_price" expr="unit_price * 0.6" />
            <elseif cond="quantity &gt; 10" />
                <assign name="unit_price" expr="unit_price * 0.8" />
            <else/>
                <assign name="unit_price" expr="unit_price" />
            </if>
            <assign name="price" expr="quantity * unit_price" />
        </block>
    </form>
</vxml>
```

```

<assign name="order_prompt" expr="order_text + price" />
<assign name="application.total_order_cost"
       expr="application.total_order_cost + price" />
<assign name="application.total_books"
       expr="application.total_books + quantity" />
</block>
</form>
</vxml>

```

这个程序清单展示了if, elseif 和else等元素的使用情况。在这个例子中, else元素不是必不可少的, 因为当下面这个语句

```
<assign name="unit_price" expr="unit_price" />
```

执行时, unit_price变量正好设置为它本身的值。之所以它包含了else元素, 目的是举例说明else的句法。

你可能已经注意到语句<if cond = “quantity > 100”>中使用了>, 而不是使用>字符。在XML中, >是个保留字符, 因此必须避开使用该字符。下面这个列表列出了XML中的保留字符及其转义符:

保留字符	转义符
< 小于符号	<
> 大于符号	>
“ 引号	"
‘ 省字号	'
& “与” 符号	&

8.10.3 嵌套的if/else/elseif语句

在VoiceXML 中, if/else/elseif 语句可以相互嵌套。你可以使用一个嵌套的if/else/elseif 语句将电子优惠条件与批量折扣组合在一起。你可以将电子优惠条件修改为只有按照全额价24.95美元购买图书时才有效, 不过你将把优惠折扣提高到30%, 请看程序清单8-7中的设置情况。

程序清单8-7 CalculateCost.vxml

```

<?xml version="1.0"?>
<!--Demonstrates the if/elseif/else elements-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" content="Sams Publishing 2001"/>
  <meta name="description"
        content="Calculate the cost of a book order, to access
        ↳call (xxx)-xxx-xxxx"/>

  <meta name="keywords" content="VoiceXML Sams Books" />
  <meta name="maintainer" content="johnnyo@sams.net" />
  <meta name="robots" content="index,follow" />
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />
  <var name="price" />

```

```

<var name="order_prompt" />
<var name="isCoupon" expr="true" />

<form id="calculate_price">
    <var name="quantity" expr="2" />
    <var name="unit_price" expr="24.95" />
    <var name="order_text" expr="The total price of the order is '' />
    <block name="assignment" >
        <if cond="quantity &gt; 100">
            <assign name="unit_price" expr="unit_price * 0.6" />
        <elseif cond="quantity &gt; 10" />
            <assign name="unit_price" expr="unit_price * 0.8" />
        <else/>
            <if cond="isCoupon == true">
                <assign name="unit_price" expr="unit_price * 0.7" />
            </if>
        </if>
        <assign name="price" expr="quantity * unit_price" />
        <assign name="order_prompt" expr="order_text + price" />
        <assign name="application.total_order_cost"
               expr="application.total_order_cost + price" />
        <assign name="application.total_books"
               expr="application.total_books + quantity" />
    </block>
</form>
</vxmle>

```

8.11 param元素

VoiceXML中的param元素与HTML中的param元素相似。它用于将参数传递给子对话框或对象。当使用param元素将参数传递给子对话框时，param属性的name以及value或expr属性均可使用。除了这些属性外，用于将参数传递给对象的param元素还具有valuetype和type属性。

value将一个值与name相关联。valuetype属性可以拥有两个值，即data或ref，其默认值是data。valuetype属性用于向对象说明与name相关的值究竟是数据（data）还是个引用号（ref）。如果valuetype是个引用号，那么type属性用于指明结果的MIME类型。

如果要了解与子对话框相关的param元素的使用方法，请参见本章后面部分中的“subdialog元素”一节的内容。object元素和param元素如何用于对象的问题，将在第10章中详细介绍。

8.11.1 name属性

name属性是传递的参数的名字。

8.11.2 expr属性

expr属性是用于计算与name相关的结果的表达式。

8.11.3 value属性

value属性用于将一个值与名字关联起来。

8.11.4 valuetype属性

valuetype属性有2个值，即data 或ref， 默认值是data。valuetype用于向对象说明与 name 相关的值究竟是数据还是引用号。

8.11.5 type属性

如果valuetype的值是个引用号， type属性用于指明结果的MIME 类型。

8.12 subdialog元素

subdialog类似函数调用。对话框调用 subdialog，以便执行特定的指令，或者收集数据。调用subdialog时既可以将执行控件传递给同一个文档中的对话框，或者传递给另一个文档中的对话框，也可以传递给使用CGI、Perl或 ASP之类的脚本编程语言的另一个文档，以便生成一个VoiceXML文档。

subdialog元素之所以非常有用，原因是它们允许程序设计人员在引用其他文档时不必退出当前的对话框，也不会丢失所有的上下文信息。另外， subdialog元素使用户可以建立相应的组件，这些组件可以在单个应用程序中的各个 VoiceXML文档之间共享，也可以被其他 VoiceXML 应用程序重复使用。

subdialog元素是在它自己的上下文环境中执行的。当return指令执行时， subdialog的执行便告完成。subdialog可以返回变量，也可以在返回时产生一个事件。当 subdialog返回时，来自调用对话框的所有本地数据、语法和状态信息均被还原。如果 subdialog 产生一个事件，便设定 subdialog变量，而subdialog的执行将在 subdialog的 filled元素中继续进行。

只有当src设定的位置上的文档是个非VoiceXML 文档，如ASP、Perl、CGI文档时，才使用 subdialog的属性 namelist 、method 和encotype 。它的其余属性caching、fetchaudio、fetchtimeout 和fetchhint均与提取资源有关，因此只有当 src属性的 URL 是另一个文档时才使用。

8.12.1 name属性

name属性是指与 subdialog相关的变量的名字。

8.12.2 src属性

src是subdialog的URL。src可以引用该文档或另一个文档中的另一个对话框， src是个必不可少的属性。

8.12.3 expr属性

如果expr属性拥有一个非undefined的值，那么就不能访问subdialog。如果 expr 属性被省略，

subdialog变量的初始值将是undefined。

8.12.4 cond属性

如果subdialog被访问，那么cond属性的值必须是true。cond属性的默认值是true。

8.12.5 modal属性

modal属性的值可以是true，也可以是false。如果是true，那么调用subdialog的对话框中有效的语法，在子对话框中也是有效的。如果modal的值是false，那么语法将是无效的。model的默认值是true。

8.12.6 namelist属性

namelist由通过http协议提交的变量列表组成。

8.12.7 method属性

method是指请求的方法，它的值可以是get也可以是post，默认值是get。

8.12.8 enctype属性

enctype是提交的文档的MIME编码类型。

8.12.9 caching属性

caching属性可以有两个值，即fast和safe。如果Caching属性设置为默认值fast，那么如果存在文档的缓存拷贝，则使用该拷贝。如果caching设置为safe，那么不管文档原先是否被平台存入高速缓存，均从URL下载文档。

8.12.10 fetchtimeout属性

fetchtimeout属性是指产生error.badfetch.event事件之前平台为检索文档而等待的间隔时间。

8.12.11 fetchhint属性

fetchhint属性用于确定平台是在需要子对话框之前开始提取包含了子对话框的文档(fetchhint=prefetch)，还是在访问subdialog元素之后开始提取该文档(fetchhint=)。fetchhint的默认值是safe。

8.12.12 fetchaudio属性

当文档下载时，fetchaudio属性的值被设置为要播放的声音文件的URL。如果没有fetchaudio属性，就不能播放任何声音文件。

8.12.13 return元素

return元素用于结束对话框的执行，并且将执行控件返回给调用子对话框的这个对话框。

event属性是子对话框返回后产生的一个事件。

namelist属性包含了子对话框完成运行后返回给调用对话框的变量的列表。

8.12.14 程序举例

在你的在线书店应用程序中，你可以建立一个子对话框，以便计算购书订单的费用，计算费用时要考虑可以享受的各种折扣。程序清单8-8显示了它的代码。

程序清单8-8 CalculateCostSub.vxml

```
<?xml version="1.0"?>
<!--CalculateCostSub.vxml-->
<!--Demonstrates the subdialog element-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>
    <meta name="description"
        content="Calculate the cost of a book order, to access
    ↳call (xxx)-xxx-xxxx"/>

    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="order_prompt" />

    <form id="calculate_price">
        <var name="order_text"
            expr="'The total price of the order is ' />
        <subdialog name="discount" src="#calculate_discount">
            <param name="quantity" expr="200" />
            <param name="unit_price" expr="24.95" />
            <param name="isCoupon" expr="true" />
            <filled>
                <assign name="order_prompt"
                    expr="order_text + discount.price" />
                <assign name="application.total_order_cost"
                    expr="application.total_order_cost +
    ↳discount.price" />
                <assign name="application.total_books"
                    expr="application.total_books + quantity" />
            </filled>
        </subdialog>
    </form>

    <form id="calculate_discount">
```

```

<var name="quantity" />
<var name="unit_price" />
<var name="isCoupon" />
<block>
    <var name="price" />
    <if cond="quantity > 100">
        <assign name="unit_price" expr="unit_price * 0.6" />
    <elseif cond="quantity > 10" />
        <assign name="unit_price" expr="unit_price * 0.8" />
    <else/>
        <if cond="isCoupon == true">
            <assign name="unit_price" expr="unit_price * 0.7" />
        </if>
    </if>
    <assign name="price" expr="quantity * unit_price" />
    <return namelist="price" />
</block>
</form>
</vxm1>

```

在这个程序清单中，窗体calculate_price 调用diacount子对话框。discount 子对话框位于窗体calculate_discount 中，而窗体calculate_discount 与调用窗体calculate_price位于同一个文档中，下面的几个代码行使用 param元素将参数 quantity, unit_price和 isCoupon 传递给子对话框：

```

<param name="quantity" expr="200" />
<param name="unit_price" expr="24.95" />
<param name="isCoupon" expr="true" />

```

窗体calculate_discount 用于根据传递的参数计算购买的商品的费用，并且通过return元素中的 namelist属性返回 price 变量。当到达return元素时，子对话框的discount变量就被赋予一个值，然后封装在filled元素中的指令便开始执行。接着，下面的语句

```
<assign name="order_prompt" expr="order_text + discount.price" />
```

被执行。请注意子对话框的名字是如何将 price变量作为一个前缀来引用的。

这个例子说明，当通过 param 元素将一个参数传递给subdialog 时，位于子对话框的src属性位置上的对话框必须声明scope对话框的各个变量。其变量名与 param元素中的变量名一致。声明的变量不能拥有已经定义的 expr 属性，因为它们是用对应的param 元素中的expr属性的值进行初始化的。

在这个程序例子中， quantity、unit_price和 isCoupon等3个变量用 src 值calculate_discount 传递给subdialog 。因此窗体calculate_discount必须并且确实使用 var 元素来声明对话框scope 的3个变量quantity, unit_price 和isCoupon 。变量quantity 的值初始化为200，这是也带有名字 quantity 的param 元素的expr属性的输出。同样， calculate_discount 窗体的unit_price的值初始化为24.95， isCoupon 的值初始化为true 。

8.12.15 另一个程序例子

子对话框也可以驻留在sbudialog被调用的文档之外的文档中，在这种情况下，你可以将窗

体 calculate_discount 移到它自己的文档中，将subdialog 元素的src属性更新为新的 URL，并且使示例程序的运行特性保持不变。在这个应用程序中你必须移动窗体calculate_discount，因为编写代码以便计算你的购物订单中的T恤衫部分的费用时，子对话框discount可以重复使用。程序清单8-9显示了更新后的应用程序。

程序清单8-9 CalculateCostSub2.vxml

```

<?xml version="1.0"?>
<!--CalculateCostSub2.vxml-->
<!--Demonstrates the subdialog element-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Calculate the cost of a book order, to access
    =>call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="order_prompt" />

    <form id="calculate_price">
        <var name="order_text"
            expr="'The total price of the order is '' '/>
        <subdialog name="discount"
            =>src=".//CalculateDiscount.vxml#calculate_discount">
            <param name="quantity" value="200" />
            <param name="unit_price" expr="24.95" />
            <param name="isCoupon" expr="true" />
            <filled>
                <assign name="order_prompt"
                    expr="order_text + discount.price" />
                <assign name="application.total_order_cost"
                    expr="application.total_order_cost +
                =>discount.price" />
                <assign name="application.total_books"
                    expr="application.total_books + quantity" />
            </filled>
        </subdialog>
    </form>
</vxml>

<?xml version="1.0"?>
<!--CalculateDiscount.vxml-->
<!--Demonstrates the subdialog element-->
<vxml version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Calculate the discount of a book order, to access

```

```

    ↳call (xxx)-xxx-xxxx" />

<meta name="keywords" content="VoiceXML Sam Books" />
<meta name="maintainer" content="johnnyo@sam.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />

<form id="calculate_discount">
    <var name="quantity" />
    <var name="unit_price" />
    <var name="isCoupon" />
    <block>
        <var name="price" />
        <if cond="quantity > 100"
            <assign name="unit_price"
                expr="unit_price * 0.6" />
        <elseif cond="quantity > 10" />
            <assign name="unit_price"
                expr="unit_price * 0.8" />
        <else/>
            <if cond="isCoupon == true">
                <assign name="unit_price" expr="unit_price * 0.7" />
            </if>
        </if>
        <assign name="price" expr="quantity * unit_price" />
        <return namelist="price" />
    </block>
</form>
</vxml>

```

在这个程序例子中，subdialog被移到了它自己的文档中，这样它就可以很容易被重复使用。窗体calculate_price调用位于calculateDiscount.vxml文件中的窗体calculate_discount的子窗体discount。src属性中使用的URL是calculateDiscount.vxml文件后随#calculate_discount的相对文件路径。calculateDiscount.vxml后随#calculate_discount的绝对路径也可以使用。Src属性中的#calculate_discount部分用于设定子窗体所在的文档calculate_discount中的窗体。本示例程序的运行特性与带有与窗体calculate_price位于同一个文档中的子窗体discount的示例程序相同。

8.13 goto元素

goto元素用于从一个对话框浏览到另一个对话框，也可以从一个窗体项转到另一个窗体项。goto元素的属性caching、fetchaudio、fetchtimeout和fetchhint均与提取资源有关，因此只适用于导航到另一个文档中的对话框。

8.13.1 next属性

next属性用于从一个对话框转到另一个对话框。它用于设定要访问的下一个对话框。如果被访问的对话框是在另一个文档中，那么窗体名必须采用<URL of next document> # <dialog>

name>这样的句法。URL可以是当前文档的绝对路径，也可以是当前文档的相对路径。如果没有提供对话框的名字，便执行URL上的VoiceXML文档中的第一个对话框，当被访问的对话框位于同一个文档中时，使用的句法是#<dialog name>。对于每个goto元素来说，next、expr、nextitem和expritem属性只能有一个属性有相应的值。

8.13.2 expr属性

expr属性的作用与next相似，差别在于expr是个表达式，其计算结果是个URL。对于每个goto元素来说，next、expr、nextitem和expritem属性只能有一个属性有相应的值。

8.13.3 nextitem属性

当goto元素用于从窗体中的一个项目导航到另一个项目时，便要使用nextitem属性，而不是使用next。Nextitem的句法与next相似，差别在于nextitem的名字不带前缀#符号，对于每个goto元素来说，next、expr、nextitem和expritem属性只能有一个属性有相应的值。

8.13.4 expritem属性

当goto元素用于从窗体中的一个项目导航到另一个项目时，必须使用属性expritem，而不是expr。Expritem属性的句法与expr相似，差别在于expritem的名字不带前缀#符号。对于每个goto元素来说，next、expr、nextitem和expritem等属性只能有一个属性有相应的值。

8.13.5 caching属性

caching属性的值有两个，即fast和safe。如果caching属性位设置为它的默认值fast，那么如果存在文档的高速缓存拷贝，便使用该拷贝。如果caching属性的值设置为safe，那么不管平台是否在以前对文档进行了高速缓存，都从URL下载该文档。

8.13.6 fetchtimeout属性

fetchtimeout是指在产生error.badfetch.event事件前平台为了检索文档而等待的间隔时间。

8.13.7 fetchhint 属性

fetchhint属性用于确定平台是在它实际需要之前就提取下一个文档(fetchhint=prefetch)，还是只有在执行goto元素时才提取下一个文档。fetchhint的默认值是safe。

8.13.8 fetchaudio属性

当文档下载时，fetchaudio属性设置为要播放的声音文件的URL。如果没有设置本属性，则不播放任何声音文件。

8.13.9 程序举例

程序清单8-10中的程序例子对Welcome.vxml示例文档进行了修改，以便显示如何使用goto

元素在文档中导航，以及如何导航到另一个文档。

程序清单8-10 WelcomeGoto.vxml

```

<?xml version="1.0"?>
<! --WelcomeGoto.vxml-->
<!--This document demonstrates the goto command -->
<vxml version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Welcome page for Sams online bookstore to access call
        ↪(xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="welcome">
        <block>
            Welcome to Sams Publishing's online bookstore.
            <goto nextitem="inventory" />
        </block>

        <block name="inventory">
            Our store allows you to conveniently purchase Voice Application
            Development with Voice XML books and t-shirts.
            <goto next="#how_many" />
        </block>
    </form>

    <form id="how_many">
        <block>
            How many books do you want?
            <goto next="CalculateCostSub2.vxml" />
        </block>
    </form>
</vxml>
```

首先，使用文语转换工具向用户读出“Welcome to Sams Publishing's online bookstore”（欢迎访问萨姆出版公司的在线书店）这行代码。接着，执行指令<goto nextitem=“inventory” />，同时程序转移到inventory程序块，在这里，向用户读出“Our store allow you to conveniently purchase Voice Application Development with VoiceXML books and t-shirts”（我们的在线商店允许你方便地购买《VoiceXML语音应用程序开发手册》和T恤衫）这行语句。然后执行<goto next=“# how_many” />这行指令，接着程序转移到how_many窗体。在这个窗体中，向用户读出“How many books do you want?”（你想买几本书？）这行语句。接着执行指令<goto next=“CalculateCostSub2.vxml” />，并且上载文档 CalculateCostSub2。由于没有设定对话框的名字，因此执行文档中的第一个对话框 calculate_price。

在这个程序例子中，可以使用 expritem来取代 nextitem 。例如，<goto expritem=“‘in’ +

‘ventory’” />相当于<goto nextitem=“inventory” />。同样，expr也可以取代expritem。这里的<goto expr=“# + ‘how-many’” />就相当于<goto ext=“#how_many” />。

8.14 clear元素

clear元素用于将变量的值设置为undefined（未定义）。

8.14.1 namelist属性

namelist属性是指要清除的变量的列表。例如，下面的语句

```
<clear namelist="total_order_cost total_books total_shirts">
```

将把变量total_order_cost、total_books和total_shirts的值设置为 undefined。

8.14.2 使用说明

清除窗体中包含的变量，会带来一个副作用，那就是该元素的提示计数器和事件计数器将被清零。这个问题将在第9章中作进一步的说明。如果要了解何时使用 clear的例子，请阅读下一节“循环”的说明。

8.15 循环

在VoiceXML中没有预定义的循环。不过你可以使用block、var、clear、if和assign等元素来模仿循环的运行特性。下面这个伪代码说明了一个重复执行5次的while循环。每次运行该循环时，counter的值就翻一番。当程序运行结束时，counter的值就变成32：

Pseudo code:

```
i=0
stop = 5
counter = 1

while i < stop do
    counter = counter * 2
    i++
end while
```

程序清单8-11中的示例程序显示了VoiceXML中实现的同一个循环。

程序清单8-11 Loops.vxml

```
<?xml version="1.0"?>
<!--Loops.vxml -->
<!--Demonstrates a while loop -->
<vxml version="1.0" >
    <form>
        <var name="stop" expr="5" />
        <var name="i" expr="0" />
        <var name="counter" expr="1" />
```

```

<block name="loop">
    <if cond="i < stop">
        <assign name="i" expr="i+1" />
        <assign name="counter" expr="counter * 2" />
        <clear namelist="loop"/>
    </if>
</block>
<!-- The loop is executed 5 times so counter is 32 --&gt;
&lt;/form&gt;
&lt;/vxml&gt;
</pre>

```

另外，if中封装的clear元素可以被goto取代，其结果相同，请参见程序清单8-12。

程序清单8-12 MoreLoops.vxml

```

<?xml version="1.0"?>
<!--MoreLoops.vxml-->
<!--Demonstrates a while loop -->
<vxml version="1.0" >
    <form>
        <var name="stop" expr="5" />
        <var name="i" expr="0" />
        <var name="counter" expr="1" />
        <block name="loop">
            <if cond="i < stop">
                <assign name="i" expr="i+1" />
                <assign name="counter" expr="counter * 2" />
                <goto nextitem="loop" />
            </if>
        </block>
        <!-- The loop is executed 5 times so counter is 32 -->
    </form>
</vxml>

```

8.16 exit元素

本章介绍的所有示例程序中，当程序运行到 `</vxml>` 标记时，文档便结束运行。使用 VoiceXML 中的 `exit` 元素，程序员就可以明确退出一个程序页，并且结束 VoiceXML 会话。当执行 `exit` 元素时，VoiceXML 文档均被下载，控制则返回给执行 VoiceXML 文档的平台。这时，平台进行什么操作，将取决于平台的运行方式，不过在大多数情况下，平台将中断电话呼叫。

8.16.1 expr属性

`expr` 属性是个计算用的表达式，其计算结果是会话的返回值。

8.16.2 namelist属性

`namelist` 属性是指退出程序时返回给平台的变量列表。

8.16.3 程序举例

程序清单8-13显示了使用exit元素的一个例子。

程序清单8-13 Exit.vxml

```
<?xml version="1.0"?>
<!--Exit.vxml-->
<!--Demonstrate the exit element-->
<vxml version="1.0">
    <form>
        <var name="status" expr="'done'" />
        <block>
            <if cond="status == 'done'">
                We are finished and will now exit.
                <exit/>
            <else/>
                We are not done so we cannot exit yet.
            </if>
                We have not exited yet.
        </block>
    </form>
</vxml>
```

在这个程序例子中，if cond属性的值应该计算为 true。用户应该首先听到文语转换（TTS）句子“*We are finished and will now exit*”（我们已经完成程序的运行，现在准备退出）。然后系统就应该退出，而不需要执行指令“*We have not exited yet*”（我们尚未退出），也不需要通过TTS向用户读出该指令。

8.17 disconnect元素

disconnect元素用于使平台切断用户的电话。

程序举例

程序清单8-14显示了使用disconnect元素的一个例子。

程序清单8-14 Disconnect.vxml

```
<?xml version="1.0"?>
<!--Disconnect.vxml-->
<!--Demonstrate the disconnect element-->
<vxml version="1.0">
    <form>
        <var name="disconnect" expr="true" />
        <block>
            <if cond="disconnect == true">
                Goodbye
                <disconnect/>
            <else/>
```

```
We are not done talking with you yet.  
</if>  
We have not yet disconnected the caller.  
</block>  
</form>  
</vxml>
```

在这个示例程序中，if cond属性的值应该计算为true。用户首先应该听到文语转换的语句“Goodbye”（再见）。然后系统应该切断用户的电话连接，而不需要通过文语转换工具向用户读出语句“*We have not yet disconnect the caller*”（我们尚未切断呼叫者的电话连接）。

disconnect元素与exit元素之间存在两个重大的差别。首先，disconnect元素能够强制平台切断用户的电话连接。而当exit元素被调用时，平台必须确定是否要切断用户的电话连接。第二，当调用exit元素时，所有VoiceXML文档的运行将全部停止。而当调用disconnect时，则能够显示telephone.disconnect.hangup事件，已经上载的VoiceXML页（或多个VoiceXML页）会抓取到该事件，并且执行必要的最后处理。

8.18 小结

本章介绍了VoiceXML的基本输入和输出命令。并且使用这些命令开始建立在线书店购物应用程序。本章介绍的元素可以分成下面几个类别，即程序结构元素，程序执行控制元素，变量操作元素，以及应用程序描述元素等。程序结构元素包括xml、vxml、block和form等。程序执行控制元素包括if、clear、param、goto、exit、return和subdialog等。变量操作元素包括var、assign、clear、param、goto、exit和disconnect等，应用程序描述元素包括meta元素。在下一章中，我们将要使用与用户进行交互操作的VoiceXML的输入和输出命令。以增强在线书店应用程序的功能。

第9章 VoiceXML的输入和输出命令

本章将要介绍通过文本语音转换（TTS）工具将信息输出给用户和播放声音文件时使用的VoiceXML元素，本章将要详细讲述如何通过语音识别和使用称为双音多频（DTMF）系统接收电话按键信息等方法来收集来自用户的输入信息。本章还要介绍如何妥善处理与输入相关的出错条件。本章详细说明了每个元素的属性和属性可以使用的值。我们将要使用本章介绍的各种标记来进一步建立语音商务站点。

9.1 VoiceXML中的输出方法

在VoiceXML中，有两种方法可以用来将信息输出给用户。第一种方法是通过文本语音转换（TTS）工具进行输出。另一种方法是通过播放预先录制的声音文件。在上一章中建立的语音商务应用程序中，“Our store allow you to conveniently purchase Voice Application Development with Voice XML books and t-shirts” 和 “How many books do you want?” 这两个TTS 输出均封装在block 标记中，请参见程序清单9-1。

程序清单9-1 WelcomeGoto.vxml

```
<?xml version="1.0"?>
<!--WelcomeGoto.vxml-->
<!--This document demonstrates the goto command -->
<vxml version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Welcome page for Sams online bookstore to access
    <call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="welcome">
        <block>
            Welcome to Sams Publishing's online bookstore.
            <goto nextitem="inventory" />
        </block>

        <block name="inventory">
            Our store allows you to conveniently purchase Voice Application
            Development with Voice XML books and t-shirts.
            <goto next="#how_many" />
        </block>
    </form>
```

```

<form id="how_many">
  <block>
    How many books do you want?
    <goto next="CalculateCostSub2.vxml" />
  </block>
</form>
</vxml>

```

block元素也可以用来封装audio元素。audio元素用于设定预先录制的声音文件，以便向用户播放。我们将在本章后面部分中的“audio元素”小节中介绍它。VoiceXML的输出可以包含声音文件与TTS的混合信息。

同时使用TTS信息和预先录制的声音文件时好处很多。使用TTS时，

- 开发人员能够非常容易地修改输出信息。
- 可以用脚本语言XSL、ASP和JSP等生成输出信息，因此它是动态的。
- 不需要增加录制声音文件的时间，就可以迅速对应用程序原型进行编码。
- 如果不需要为专业人员支付录制声音文件的费用，那么就可以降低应用程序的开发费用。

如果使用声音文件，那么：

- 用户对应用程序就会有一个更好的感觉，因为几乎在所有情况下，用户都希望接收声音文件的输出，而不希望接收TTS信息。
- 开发人员不必花时间给输出添加语言标记，以便使它听起来“恰好正确”

在本章的下面各节和下面各章中的示例代码中，凡是增加的文本输出都是使用TTS给出的，因为这使得代码比较容易阅读。

9.1.1 prompt元素

prompt元素是封装TTS输出和audio元素时使用的另一个元素。当TTS信息不用prompt元素封装时，便可推定prompt元素使用了默认属性。

在两种情况下，必须用prompt元素来封装输出。首先，如果输出包含TTS和语音标记元素，就必须使用prompt元素。语音标记元素用于控制向用户说出TTS信息时的各项特性，如速度，音量和发音等。（详细说明请参见本章后面部分中的“语音标记”这一节的内容。）第二种情况是，当输出信息需要bargein、cond、count或timeout等属性时，就必须使用prompt元素：

- bargein——如果bargein属性的值是true，用户可以使用语音应答或DTMF来中断输出。如果bargein的值是false，用户将不能中断输出，输出将一直播放下去，直到结束。
- cond——cond属性使用的是布尔值，用于确定究竟是向用户输出文本还是声音文件。它的默认值是true。
- count——当首次访问菜单或窗体项时，它的初始提示计数器的值被初始化为1。只有当count的值等于提示计数器的值，或者小于提示计数器的值但是拥有菜单或窗体项中的任何提示的最大值时，才输出提示。当一个提示被选定并且排队等候播放之后，提示计数器的值便递增1，当退出菜单或窗体项时，提示计数器便清零。
- timeout——这是提示播放后，产生noinput事件之前，平台等待用户作出应答的时间。

timeout的默认值是各种平台根据自己的情况设定的。

在本章的“field元素”这一节中，你可以找到bargein和timeout属性的例子。在本章的“事件”一节中可以看到count属性的例子。程序清单9-2中的示例代码使用带有cond属性的prompt元素，建立一个新窗体，以便通知用户说他采购的商品已经添加给他的购物车。

程序清单9-2 CalculateCostPrompt.vxml

```
<?xml version="1.0"?>
<!--CalculateCostPrompt.vxml-->
<!--Demonstrates the prompt element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001" />

    <meta name="description"
          content="Calculate the cost of a book order,
          to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
    <meta name="robots" content="index,follow" />
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST" />
    <var name="order_prompt" />
    <var name="number_of_books" expr="200" />

    <form id="calculate_price">
        <var name="order_text"
            expr="'The total price of the order is ' " />
        <subdialog name="discount"
                  src=".//CalculateDiscount.vxml#calculate_discount">
            <param name="quantity" expr="number_of_books" />
            <param name="unit_price" expr="24.95" />
            <param name="isCoupon" expr="true" />
            <filled>
                <assign name="order_prompt"
                       expr="order_text + discount.price" />
                <assign name="application.total_order_cost"
                       expr="application.total_order_cost + discount.price" />
                <assign name="application.total_books"
                       expr="application.total_books + number_of_books" />
                <goto next="#output_total" />
            </filled>
        </subdialog>
    </form>

    <form id="output_total">
        <block>
            Your
            <prompt cond="number_of_books == 1">book has</prompt>
            <prompt cond="number_of_books != 1">books have</prompt>
        </block>
    </form>
```

```

    been added to your shopping cart.
  </block>
</form>
</vxml>
```

当output-total 窗体被访问时，首先向用户读出文字“Your”，接着下面的指令

```
<prompt cond="number_of_books == 1">book has</prompt>
```

被执行。由于number_of_book 的值等于200，cond属性的值计算为false，因此提示不读出。接着，下面这条指令

```
<prompt cond="number_of_books > 1">books have</prompt>
```

被执行。由于number_of_books的值大于1，因此向用户读出“books have”（图书已经）。最后输出的是“been add to you shopping cart”（被添加到你的购物车中）。

9.1.2 语音标记

语音标记是赋予 sayas、break、div、emp和 pros 等元素的名字。这些元素总是被嵌入 prompt元素中，语音标记可用于调节如何向用户播放TTS信息的声音。它采用的是 Java API语音标记语言技术标准（0.5版，1997年8月 28日推出）。详细信息请访问网址http://java.sun. com/products/java-media/speech/forDevelopers/JSQL。

有些VoiceXML 平台只支持一部分语音标记元素和属性，原因是平台使用的TTS 服务器受到硬件和软件的限制。如果某个平台不支持某个语音标记元素，它就应该忽略该元素，并且按正常方式输出TTS信息。

程序清单9-3中的示例应用程序将语音标记元素纳入了你的语音商务应用程序。使用户能够更容易理解TTS输出。这个示例应用程序显示了语音标记元素是如何嵌套的。

程序清单9-3 WelcomeSM.vxml

```

<?xml version="1.0"?>
<!--WelcomeSM.vxml-->
<!--This document demonstrates the speech markup-->
<vxml version="1.0" application="BookStoreRoot">
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" content="Sams Publishing 2001"/>

  <meta name="description"
        content="Welcome page for Sams online bookstore,
        to access call (xxx)-xxx-xxxx"/>

  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <form id="welcome">
    <block>
      <prompt>
        <pros vol="+20%">
```

```

<div type="paragraph">
    <emp level="strong">Welcome</emp>
    to Sams Publishing's online bookstore.

    Our store allows you to conveniently purchase Voice
    Application Development with Voice XML books and
    t-shirts. You can also visit us on the web at
    <break msecs="100" />
    <sayas sub="W W W dot voice genie dot com ">
        www.voicegenie.com
    </sayas>
</div>
</pros>
</prompt>
<goto next="#how_many" />
</block>
</form>

<form id="how_many">
<block>
    How many books do you want?
    <goto next="CalculateCostPrompt.vxml" />
</block>
</form>
</vxml>

```

由于TTS的播放过程很长，因此用户很难收听和完全理解传递给他们的所有信息。语音标记提供了一种方法，通过插入一些停顿，调整某些单词的发音，并且改变说话的音量和速度，使得很长的TTS播放过程看上去很象是自然的对话。不过没有一种固定的方法可以用来选择语音标记元素的正确值及其属性值。开发人员常常采取试用不同的语音标记值和了解它的错误值的方法，找出“最佳”的输出。

1. sayas元素

sayas元素用于设定单词或词组应该如何发音。它配下列属性：

- phon——phon属性使开发人员能够在应用程序这一级上控制某些单词应该如何发音。Phon使用国际音标（IPA）的Unicode字符集来定义一系列的发音。Java API语音标记技术标准称，“IPA字符由 \u0250至\u02AF的代码、\u02B0至\u02FF的修饰符、\u0300至\u036F的读音符号、某些拉丁文和希腊文、以及\u0000至\u017F这个范围的符号字符来代表。Unicode IPA 支持特性的详细说明请参见Uincode标准第2.0版（ Unicode集团编制，Addison-Wesley Developers出版社1996年出版）。该技术标准还提供了一个例子，在这个例子中，单词 phonetitecs的发音使用了一系列的 unicode声音：

```
<sayas phone="\u0066\u006F\u028A\u006E\u025B\u0074\u026A\u006B\u0073">
    phonetics
</sayas>
```

- sub——sub属性用于定义说出的替代文字而不是实际包含的文字。
- class——class属性的有效值是 phone, date, digit, literal, currency, number 和time。

class属性用于通知平台，封装的文本应该用特定的格式来输出。

与sayas元素相同，value元素也有一个class属性。如果要了解class属性的举例，请参见本章的“value元素”一节的内容。

在WelcomeSM.vxml程序中，语句

```
<sayas sub="W W W dot voice genie dot com ">www.voicegenie.com</sayas>
```

将VoiceXML平台设定为字符串“www.voicegenie.com”应该用Web地址的约定来发音。

2. break元素

break元素用于设定语音输出的一个停顿。本元素配有下列属性：

- msecs——本属性是指停顿的毫秒数。
- size——size属性用于描述停顿的持续时间。它的值可以是none, small, medium和large。none, small, medium和large的持续时间要根据不同的平台而定。

无论是msecs和size属性，都可以用于break元素，但是两者不能同时使用。如果这两个属性都没有设定，则假设size=“medium”。

在WelcomeSM.vxml示例程序中，break元素`<break msecs="100" />`位于提示中的web地址前：

```
<prompt>
  <pros vol="+20%">
    <div type="paragraph">
      <emp level="strong">Welcome</emp>
      to Sams Publishing's online bookstore. Our store allows you to
      conveniently purchase Voice Application Development with
      Voice XML books and t-shirts. You can also visit us on the web
      at <break msecs="100" />

      <sayas sub="W W W dot voice genie dot com ">www.voicegenie.com
    </sayas>
    </div>
  </pros>
</prompt>
```

在程序段落的这个位置上应该有一个停顿，因为它将使用户有机会准备写下web地址。虽然TTS引擎能够在句号和逗号位置上自动停顿，但是当阅读很长的信息时，有时必须设置一些停顿，使用户有额外的时间明智地处理信息。

3. div元素

div元素用于确定任何封装的TTS将以句子或段落形式作为平台的输出。它配有下面这个属性：

- type——type属性用于确定平台将如何输出TTS信息。Type属性的值可以是sentence(句子)，也可以是paragraph(段落)。

在WelcomeSM.vxml示例程序中，`<div type="paragraph">...</div>`被封装在一个提示中：

```
<prompt>
  <pros vol="+20%">
```

```

<div type="paragraph">
    <emp level="strong">Welcome</emp>
    to Sams Publishing's online bookstore. Our store allows you to
    conveniently purchase Voice Application Development with Voice
    XML books and t-shirts. You can also visit us on the web at
    <break msecs="100" />
    <sayas sub="W W W dot voice genie dot com ">
        www.voicegenie.com
    </sayas>
</div>
</pros>
</prompt>

```

<div type=“paragraph”>...</div>这个指令告诉平台说，封装的文本是个段落，它应该正确地读出。

4. emp元素

emp元素用于设定封装的TTS信息说出时应该使用什么样的强调等级。它配有下面这个属性：

- level——level属性用于设定单词的强调等级。它的值可以是strong, moderate, none和reduced。默认值是moderate。

在WelcomeSM.vxml示例程序中，<emp level=“strong”> Welcome </emp>用于指明单词“welcome”应该用特定的强调语气来发音。以便强调说明该应用程序希望用户感到欢迎他使用该应用程序。

5. pros元素

pros元素用于设定说出封装的 TTS信息时的速度，音量，音调和音调范围。它配有下列属性：

- rate——rate属性用于设定说出 TTS信息时所用的速度。
- vol——vol属性用于设定说出 TTS信息时所用的音量。
- pitch——pitch属性用于设定说出 TTS信息时的所用的音调。
- range——range属性用于说出 TTS 信息时所用的音调范围。

在WelcomeSM.vxml示例程序中，<pros vol = “+20%”> ...</pros>封装在一个指示中：

```

<prompt>
    <pros vol="+20%">
        <div type="paragraph">
            <emp level="strong">Welcome</emp>
            to Sams Publishing's online bookstore. Our store allows you to
            conveniently purchase Voice Application Development with Voice
            XML books and t-shirts. You can also visit us on the web at
            <break msecs="100" />
            <sayas sub="W W W dot voice genie dot com ">
                www.voicegenie.com
            </sayas>
        </div>
    </pros>
</prompt>

```

<pros vol = “+20%” > ...</pros>这个指令告诉平台说，封装的文字播放时的音量应该比前面的音量水平高20%。在你的示例程序中，前面的音量水平是平台的默认音量水平。

9.1.3 value元素

value元素用于输出变量的值和表达式的计算结果。变量既可以对应于录制的声音文件，也可以对应于某种变量的类型，比如 undefined、null、boolean、string、number或 object。当输出的值是个录制的声音文件时，该声音文件将被播放给用户。否则，变量或表达式的计算结果便通过TTS输出给用户。

Value元素配有下列属性：

- expr——expr属性是计算的表达式，它的计算结果将被输出给用户。
- class——class属性的有效值是 phone、date、digits、literal、currency、number和time。class属性用于将封装的文本输出时采用的特定格式通知平台。
- mode——mode属性的值有两个，即默认值 tts（文语转换）和recorded。如果输出的值是一个录制的声音文件，mode应该是recorded，否则，mode应是 tts。
- recSrc——recSrc是当 mode是 recorded时要连接的声音文件的 URL。

9.1.4 程序举例

下面让我们再次访问示例程序CalculateCostPrompt.vxml，当用户将他要购买的图书添加到他的购物车上去时，该程序将输出一个确认消息。在下面的程序清单9-4中所示的程序中，你使用value元素来扩展确认消息，使之包含添加给购物车的图书数量、这些图书的费用、购物车中的图书和T恤衫的合计数量，以及购物车中的所有商品的合计费用。

程序清单9-4 CalculateCostValue.vxml

```

<?xml version="1.0"?>
<!--Root document -->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>

    <meta name="description"
          content="Root document for our book store,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />

```

```

</vxm1>

<?xml version="1.0"?>
<!--CalculateCostValue.vxml-->
<!--Demonstrates the value element -->
<vxm1 version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>

    <meta name="description"
        content="Calculate the cost of a book order,
            to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>
    <var name="added_cost" />
    <var name="added_books" expr="200" />

    <form id="calculate_price">
        <block>
            <assign name="application.total_books"
                expr="application.total_books + added_books" />
        </block>
        <subdialog name="discount"
            src=". /CalculateDiscount.vxml#calculate_discount">
            <param name="quantity" expr="application.total_books" />
            <param name="unit_price" expr="24.95" />
            <param name="isCoupon" expr="false" />
            <filled>
                <assign name="application.total_book_cost"
                    expr="Number(discount.price)" />

                <!-- Keep track of the cost of the books added -->
                <var name="old_cost"
                    expr="application.total_order_cost" />
                <!-- Cost of the books added = new total - old total. -->
                <assign name="application.total_order_cost"
                    expr="application.total_book_cost
                        + application.total_tshirt_cost" />
                <assign name="added_cost"
                    expr="application.total_order_cost - old_cost" />
                <goto next="#output_total" />
            </filled>
        </subdialog>
    </form>

    <form id="output_total">
        <block>

```

```

You have added <value expr="String(Math.round(added_books))" />

<prompt cond="added_books == 1">book</prompt>
<prompt cond="added_books != 1">books</prompt>

to your shopping cart. The cost of the
<value expr="String(Math.round(added_books))" /> added
<prompt cond="added_books == 1">book</prompt>
<prompt cond="added_books != 1">books</prompt>

is <value expr="added_cost" class="currency" />. Your
shopping cart now contains
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">book</prompt>
<prompt cond="application.total_books != 1">books</prompt>

and
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">t-shirt</prompt>
<prompt cond="application.total_shirts != 1">t-shirts</prompt>

The total cost for your shopping cart is
<value expr="application.total_order_cost" class="currency" />
</block>
</form>
</vxml>

```

在这个示例程序中，变量 `total_book_cost` 和 `total_tshirt_cost` 被添加给文档 `BooksStoreRoot.vxml`。`total_book_cost` 是购物车中目前所有图书的费用。`Total_tshirt_cost` 是购物车中目前所有T恤衫的费用。

`value` 元素用于输出变量 `added_books`、`added_cost`、`total_books`、`total_costs` 和 `total_order_cost` 的内容。`value` 元素被嵌入 `block` 元素中，并且混杂在 TTS 字符串中。`value` 元素也可以封装在 `prompt` 元素之中，并且如你在“audio 元素”这一节的示例程序中看到的那样，`value` 元素也可以混杂在预先录制的声音文件中，你可能已经注意到 `order_text` 和 `order_prompt` 变量已经从下面这个示例程序中删除了：

```

<var name="order_prompt" />

<form id="calculate_price">
<var name="order_text"
      expr="'The total price of the order is ' />
<subdialog name="discount"
           src=".//CalculateDiscount.vxml#calculate_discount">
<param name="quantity" value="200" />
<param name="unit_price" expr="24.95" />
<param name="isCoupon" expr="true" />
<filled>
    <assign name="order_prompt"

```

```
expr="order_text + discount.price" />
```

这个程序中的图书费用与 order_text 中包含的文本是分开输出的。通过分开输出图书的费用，你就能够利用 value 元素中的 class 属性的各项特性。通过将 currency 值赋予 class 属性，你就可以将变量 total_order_cost 和 added_cost 作为货币值而不是十进制数字来输出。

在输出 added_books、total_books 和 total_shirts 时使用的 value 语句中，每个变量都有下面的类型代码所封装

```
String(Math.round(<variable>))
```

这个封装文本包含 ECMAScript 函数 Math.round() 和 string()。执行这两个函数的结果是可以确保被它封装的变量将用一个整数值来说出。例如，在示例程序 CalculateCostValue.vxml 中，当下面的指令

```
<value expr="total_books" />
```

执行时，变量 total_books 将被说成是“two-hundred point zero”。当下面的指令

```
<value expr="String(Math.round(<variable>))" />
```

执行时，变量 total_books 就被说成是“two-hundred”。之所以需要使用中这些函数，目的是确保变量 added_books、total_books 和 total_shirts 能够正确输出，因为这些变量均属于 Number 类型的变量。因此 VoiceXML 平台可能将这些变量的值作为十进制数字输出。包括 String() 和 Math.round() 在内的 ECMAScript 函数将在第 11 章中进一步介绍。

为了输出购买图书的合计费用，你必须更改 filled 元素中封装的逻辑，以便计算图书的合计费用和购物车中的商品的合计费用。现在购物车中所有图书的费用将由子对话框 discount 来计算。由于子对话框返回的变量类型是 object，因此子对话框 discount 的 namelist 返回的变量 price 是变量类型 object。下面的语句

```
<assign name="application.total_book_cost" expr="Number(discount.price)" />
```

使用 price 的变量类型 object，再将它转换成变量类型 Number，然后将变量类型 Number 赋予 total_book_cost。

当变量 total_book_cost 与 total_tshirt_cost 相加时，它必须属于变量类型 Number，这样才能得出购物中商品的新费用的正确结果。如果下面语句中的变量 total_book_cost 的类型是 Object

```
<assign name="application.total_order_cost" expr="application.total_book_cost + application.total_tshirt_cost" />
```

那么当命令执行时，变量 total_book_cost 和 total_tshirt_cost 将被并置，而不是相加，得出的结果将是 29940，而不是 2994。最后，存放在变量 old_cost 中的加上图书费用前购物车中的商品合计费用，将要从存放在变量 total_order_cost 中的购物车商品的当前合计费用中减去。结果得出添加给购物车的图书的费用，它存放在变量 added_cost 中。

本示例程序中的所有 value 元素都没有设定 mode 的值，也就是说 mode 使用的是它的默认值 tts。对于本示例程序中的每个 value 元素来说，value 元素的 expr 属性的计算结果将是使用 TTS 的输出。在第 10 章的“record 元素”这一节中，可以看到引用录制的声音文件并且将 mode 设置为 recorded 的变量的例子。

9.1.5 audio元素

录制的声音文件和TTS是VoiceXML中将信息输出给用户的两种方法。audio元素用于设定向用户播放预先录制的声音文件。不同的VoiceXML平台支持不同的声音文件格式。最常用的声音文件类型是.wav文件，不过你必须查看VoiceXML平台的文档，才能了解它支持的声音文件格式的列表。

audio元素配有下列属性：

- src——src属性是向用户播放的声音文件的URL。
- caching——caching属性可以使用fast和safe这两个值中的一个。如果caching设置为它的默认值fast，那么如果有任何一个声音文件的高速缓存拷贝，并且它没有过期，那么就使用该拷贝。如果caching设置为safe，那么不管平台以前是否将声音文件放入高速缓存，都要从URL那里下载该声音文件。Caching的值通常设置为fast，即预先录制的声音文件，因为当应用程序设计完成并且部署后，该文件不会经常变更。
- fetchtimeout——本属性用于设定出现提取声音文件超时之前，平台为检索声音文件而等待的时间间隔。如果出现了提取声音文件的超时，并且audio元素封装的文本像本节的示例程序中的情况一样，那么该文本将输出该用户。如果出现了提取声音文件的超时，并且audio元素封装了另一个audio元素，那么平台接下来将设法下载被封装的audio元素中引用的预录制的声音文件。例如，如果在试图下载下面的语句中的声音文件fife.wav时出现了提取声音文件的超时

```
<audio src="file.wav" fetchtimeout="3s">
  <audio src="fetchtimeoutfile.wav"></audio>
</audio>
```

那么该平台接下来将设法下载文件fetchtimeoutfile.wav。如果audio元素没有封装任何文字或audio元素

```
<audio src="file.wav" fetchtimeout="3s" />
```

那么该平台将发出一个出错事件。Fetchtimeout的默认值是由不同的平台自己设定的。

- fetchhint——本属性用于确定平台是否应该在实际需要前预先提取声音文件。如果fetchhint属性的值是safe，那么只有当需要时才下载声音文件。如果fetchhint属性的值是默认值prefetch，那么平台将在上载web页时下载声音文件。如果fetchhint的值是stream，那么平台将不等到声音文件下载完毕，便开始播放该文件。

9.1.6 使用举例

程序清单9-5中的程序用预先录制的声音文件取代了不是通过value元素生成的所有TTS。

程序清单9-5. welcomeAudio.vxml

```
<?xml version="1.0"?>
<!--WelcomeAudio.vxml-->
<!--This document demonstrates the audio command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
```

```
<meta name="author" content="John O'Reilly" />
<meta name="copyright" content="Sams Publishing 2001" />

<meta name="description"
      content="Welcome page for Sams online bookstore,
      to access call (xxx)-xxx-xxxx" />

<meta name="keywords" content="VoiceXML Sams Books" />
<meta name="maintainer" content="johnnyo@sams.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />
<form id="welcome">
  <block>
    <prompt>
      <audio src="welcome.wav" caching="fast" fetchtimeout="4s"
             fetchhint="prefetch">
        <pros vol="+20%">
          <div type="paragraph">
            <emp level="strong">Welcome</emp>
            to Sams Publishing's online bookstore.
            Our store allows you to conveniently purchase
            Voice Application Development with Voice XML
            books and t-shirts. You can also visit us on the
            web at <break msecs="100" />
            <sayas sub="W W W dot voice genie dot com ">
              www.voicegenie.com
            </sayas>
          </div>
        </pros>
      </audio>
    </prompt>
    <goto next="CalculateCostAudio.vxml" fetchaudio="fetchaudio.wav" />
  </block>
</form>
</vxml>

<?xml version="1.0"?>
<!--CalculateCostAudio.vxml-->
<!--Demonstrates the mixed audio files and text-to-speech element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly" />
  <meta name="copyright"
        content="Sams Publishing 2001" />

  <meta name="description"
        content="Calculate the cost of a book order,
        to access call (xxx)-xxx-xxxx" />
  <meta name="keywords" content="VoiceXML Sams Books" />
  <meta name="maintainer" content="johnnyo@sams.net" />
  <meta name="robots" content="index,follow" />
  <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST" />
  <meta name="maintainer" content="johnoreilly@lucent.com" />
```

```

<meta name="application" content="Book Store"/>
<var name="added_cost" />
<var name="added_books" expr="200" />

<form id="calculate_price">
    <block>
        <assign name="application.total_books"
               expr="application.total_books + added_books" />
    </block>
    <subdialog name="discount"
               src=".//CalculateDiscount.vxml#calculate_discount"
               fetchaudio="fetchaudio.wav">
        <param name="quantity" expr="application.total_books" />
        <param name="unit_price" expr="24.95" />
        <param name="isCoupon" expr="false" />
        <filled>
            <assign name="application.total_book_cost"
                   expr="Number(discount.price)" />

            <!-- Keep track of the cost of the books added -->
            <var name="old_cost"
                 expr="application.total_order_cost" />
            <!-- Cost of the books added = new total - old total. -->
            <assign name="application.total_order_cost"
                   expr="application.total_book_cost +
                         application.total_tshirt_cost"/>
            <assign name="added_cost"
                   expr="application.total_order_cost - old_cost" />
            <goto next="#output_total" />
        </filled>
    </subdialog>
</form>

<form id="output_total">
    <block>
        <audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
        <audio src="added.wav" fetchtimeout="2s"> added </audio>
        <value expr="String(Math.round(added_books))" />

        <prompt cond="added_books == 1">
            <audio src="book.wav" fetchtimeout="2s">book</audio>
        </prompt>
        <prompt cond="added_books != 1">
            <audio src="books.wav" fetchtimeout="2s">books</audio>
        </prompt>

        <audio src="to.wav" fetchtimeout="2s">to </audio>
        <audio src="yourcart.wav" fetchtimeout="2s">
            your shopping cart.
        </audio>
        <audio src="thecost.wav" fetchtimeout="2s">
            The cost of the
        </audio>
    </block>
</form>

```

```

</audio>
<value expr="String(Math.round(added_books))"/>
<audio src="added.wav" fetchtimeout="2s">added</audio>

<prompt cond="added_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="added_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="is.wav" fetchtimeout="2s">is </audio>
<value expr="added_cost" class="currency"/>.
<audio src="cartcontains.wav" fetchtimeout="2s">
    Your shopping cart now contains
</audio>
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="application.total_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="and.wav" fetchtimeout="2s"> and </audio>
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">
    <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
</prompt>
<prompt cond="application.total_shirts != 1">
    <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
</prompt>

<audio src="totalcost.wav" fetchtimeout="2s">
    The total cost of the items in your shopping cart is
</audio>
<value expr="application.total_order_cost" class="currency"/>
</block>
</form>
</vxml>

```

在文档WelcomeAudio.vxml中，下面这个指令：

```

<audio src="welcome.wav" caching="safe" fetchtimeout="2s" fetchhint="prefetch">
<pros vol="+20%">
    <div type="paragraph">
        <emp level="strong">Welcome</emp>
        to Sam's Publishing's online bookstore.
        Our store allows you to conveniently purchase Voice
        Application Development with Voice XML books and t-shirts.

```

```

    You can also visit us on the web at <break msecs="100" />
    <says sub="W W W dot voice genie dot com ">
        www.voicegenie.com
    </says>
</div>
</pros>
</audio>

```

用于通过audio元素的src属性来设定一个要播放的预先录制的声音文件，名字为welcome.wav。在这个示例代码中，welcome.wav驻留在与welcomeAudio.vxml相同的目录中，但是你可以非常容易地将welcome.wav放在Web服务器上，并且设定它的URL。

当初次遇到src等于welcome.wav的audio元素时，便从Web服务器那里下载该声音文件，然后由VoiceXML平台播放。通过fetchhint属性赋予的prefetch值，你就可以让VoiceXML平台在执行audio元素之前下载声音文件welcome.wav。预先提取声音文件可以使声音文件更快地下载。

声音文件下载时，voiceXML平台也会将welcome.wav文件的一个拷贝放在平台上，当声音文件放入高速缓存后，再次要求播放welcome.wav文件并且caching属性是默认值fast时，如果caching属性还没有过期，平台就不必再次下载welcome.wav文件，而是只需要播放存储在平台上的这个文件拷贝就行了。

welcome.wav取代的文本被封装在audio元素中，如果不能在fetchtimeout属性规定的2秒钟内下载声音文件，或者你在不支持.wav文件格式的平台上运行该文档，或者你因为任何其他原因而下载或播放声音文件，那么将播放audio元素中封装的文本。VoiceXML允许在audio元素中封装语言标记、value元素、声音文件元素和混合声音文件，以及TTS。

在示例程序CalculateCostAudio.vxml中，audio元素将以下面的形式相继出现：

```

<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
<audio src="added.wav" fetchtimeout="2s"> added </audio>

```

当audio元素相继出现时，它们将在一个声音文件之后立即播放另一个声音文件，如果录制正确的话，人们就会感到这两个声音文件录制成了一个声音文件一样。

声音文件也可以与TTS信息混合在一起。当下面的命令：

```

<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
<audio src="added.wav" fetchtimeout="2s"> added </audio>
<value expr="String(Math.round(added_books))" />

```

执行时，首先向用户播放“You have added”（你已经增加），然后通过TTS向用户说出“two hundred”（200）。

audio元素既可以是block元素的子元素，也可以是prompt元素的子元素。在CalculateCostAudio.vxml程序文件中，下面的audio元素：

```
<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
```

是block元素的子元素。在同一个程序文件中，下面的audio元素：

```
<audio src="book.wav" fetchtimeout="2s">book</audio>
```

是prompt元素的子元素。

程序文件 WelcomeAudio.vxml 和 CalculateCostAudio.vxml 都提供了使用fetchaudio属性的例子。在WelcomeAudio.vxml中，下面的指令：

```
<goto next="CalculateCostAudio.vxml" fetchaudio="fetchaudio.wav" />
```

用于设定当下载 CalculateCostAudio.vxml 文件时，VoiceXML平台将播放fetchaudio.wav 文件。而在 CalculateCostAudio.vxml 文件中，下面的指令：

```
<subdialog name="discount" src=".//CalculateDiscount.vxml#calculate_discount"
fetchaudio="fetchaudio.wav">
```

用于设定下载 CalculateDiscount.vxml 文件时，VoiceXML平台将播放 fetchaudio.wav 文件，如果你一面等待文件下载，一面又播放声音文件，就可以减少用户必须经过的“停播时间”量。

9.2 接收用户的输入

到现在为止，本章介绍的内容均与向用户输出数据有关。下面各节内容将集中介绍如何接收用户输入的信息。输入信息可以用两种方式来接收，一种是语音，一种是双音多频（DTMF）。输入信息既可以直接识别，也可以录音。直接识别的输入信息是通过语法来表述的，该语法设定了一组声音，用户可以发出这一组声音，也可以由用户按下DTMF按钮，以便要求进行某项操作，或者将信息提供给应用程序。

9.2.1 field元素

field元素用于接收来自用户的自动语音识别信息或DTMF信息。它配有下列属性：

- name——name是用于存放来自用户的输入信息的变量。
- expr——expr是 name 属性中声明的变量的初始值。expr 并不是个必不可少的属性，如果它被省略了，那么该变量的值将是 undefined。如果变量的值是 undefined 之外的某个值，那么封装在field元素中的指令将不予执行。
- cond——cond是个表达式，它计算的结果可以是 true，也可以是false。如果 cond 的值计算为true，那么封装在field元素中的指令将被执行，否则便跳过 field 封装的指令。
- type——type用于指明期望的用户输入信息的类型。内置的类型是 boolean、date、digits、currency、number、phone 和 time。type属性是可有可无的。如果没有设定type，field变量便与 field内声明的用户定义的语法相关联。
- boolean——当用户说“yes”或者按下DTMF键1时，便返回 true 值。当用户说“no”或者按下DTMF键2时，便返回 false值。

Yes和no的默认DTMF按键可以被取代，下面的代码将取代 yes 的默认DTMF值：

```
<field type="boolean?y=7" />
```

这个代码中yes 的DTMF值被设置为7，同样， no的 DTMF值可以设置为8。请看下面的代码：

```
<field type="boolean?n=8" />
```

另外，yes和no的DTMF值都可以进行更换，请看下面的代码：

```
<field type="boolean?y=7;n=8" />
```

- date——有效的应答应该包含一个由月、日和年份组成的有效日期。口述的日期，比如“July 4, 1976”将用yyyymmdd格式保存为19760704。有效的DTMF应答将包含8位数日期，格式为yyyymmdd。

- digits——有效的应答是一串0到9的数字，可以口述，也可以作为DTMF来输入。词组“two hundred twenty-one”是无效的，但是词组“two two one”是有效的。

你可以为一串数字设定一个长度、最小长度和最大长度。若要设定n的长度，请使用下面的代码：

```
<field type="digits?length=n" />
```

若要设定n的最小长度，请使用下面的代码：

```
<field type="digits?minlength=n" />
```

若要设定n的最大长度，请使用下面的代码：

```
<field type="digits?maxlength=n" />
```

若要设定x的最小长度和y的最大长度，请使用下面的代码：

```
<field type="digits?minlength=x;maxlength=y" />
```

- currency——有效的应答由一个货币数量组成。如果是DTMF输入，货币的输入格式的uuudd*dd，其中uuu是ISO标准4217：1995中规定的3位数货币代码，星号是指小数点。

- number——有效的语音应答包含的数字，比如“two-hundred twenty-one”，小数点用单词“point”表示。如果是DTMF输入，输入的数字是正数，星号代表小数点。

- phone——有效的语音应答包含一串构成电话号码的数字。当输入是DTMF时，星号表示后面的数字是分机号码。例如，如果输入的数字是“8008642396*5472”，那么电话号码是800-864-2396，分机号是5472。

- time——有效的应答将包含一个由小时和分钟组成的有效时间。保存日期的格式是hhmmx，其中hh是小时，mm是分钟，x是指“a”，即AM（上午），“p”是PM（下午），“h”是用24小时的时钟表示的时间，“?”是指一个不确定的模糊时间。语音说出的时间，比如“four thirty PM”将保存为0430p。DTMF输入的时间采用hhmm格式。使用DTMF输入时，如果hh是0或者13-23，那么平台将指定“h”所属的x，否则使用“?”。

- slot——slot只用于识别语法时返回多个值的平台。在这种情况下，slot用于确定返回的值中哪一个值将赋予name。

- modal——如果modal是true，以前有效的所有语法将被关闭，只有field中声明的语法才有效。

1. field 影子变量

field元素使用下列影子变量：

- name \$.confidence——confidence是0.0-1.0这个范围中的一个数字。0.0这个值代表返回的语法或该字段中的数字可信度很低。1.0这个值表示返回的语法或该字段的数字可信度很高。
- name \$.utterance——utterance是被识别的这个确切词组。例如，如果语法包含了词汇“tee shirt”供你选择，你将“tee”定为可选单词，这样，“tee shirt”和“shirt”都是有效应答，utterance将抓取用户响应的某个有效词组。

- name \$.inputmode——inputmode（输入方式）可以是 dtmf，也可以是 voice，这要根据用户是通过语音还是输入DTMF来作为应答而定。

2. 程序举例

在前面的示例程序中，添加给购物车的图书数量以及用户是否可以享受优惠条件，已经分别被编码为200和false。在程序清单9-6这个示例程序中，你允许用户输入要添加的图书数量和他是否可以享受优惠条件，从而使该应用程序更加实用。程序GetCouponInfo.vxml询问用户他是否有优惠条件。如果用户的回答是“yes”，便输入优惠券号并进行核实。PurchaseStuffField.vxml接收用户添加给购物车的图书数量。CalculateCostField.vxml在计算图书费用时使用该信息进行必要的修改。

程序清单9-6 BookStoreRoot.vxml, GetCouponInfo.vxml, PurchaseStuffField.vxml和CalculateCostField.vxml

```

<?xml version="1.0"?>
<!--Root document -->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>

    <meta name="description"
        content="Root document for our book store, to access
    =>call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sam Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />
    <var name="num_current_order" expr="0" />
    <var name="has_coupon" expr="false" />
    <var name="asked_about_coupon" expr="false" />
</vxml>

<?xml version="1.0"?>
<!--GetCouponInfo.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sam's Publishing 2001"/>
    <meta name="description"
        content="Get coupon from user if he or she has one,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>

```

```

<meta name="robots" content="index,follow" />
<meta http-equiv="Date"
      content="Thu, 7 Aug 2000 20:20:20 EST"/>
<meta name="maintainer" content="johnoreilly@lucent.com"/>
<meta name="application" content="Book Store"/>

<form id="ask_for_coupon">
    <field name="whether_coupon" type="boolean?y=1;n=0">
        <prompt>
            Do you have a coupon?
        </prompt>
        <filled>
            <assign name="application.asked_about_coupon"
                   expr="true" />
            <if cond="whether_coupon==false" >
                <!-- if the user does not have a coupon,
                    just transition to the add x items to shopping
                    cart page. -->
                <goto next="PurchaseStuffField.vxml" />
            </if>
        </filled>
    </field>

    <field name="coupon_number" type="digits?length=5" >
        <prompt>
            Please enter your five digit coupon number?
        </prompt>
        <filled>
            <!-- For now, all valid coupons will have the
                same number -->
            <if cond="coupon_number=='55555'">
                <assign name="application.has_coupon" expr="true" />
                <goto next="PurchaseStuffField.vxml" />
            </if>
        </filled>
    </field>

    <field name="try_again" type="boolean?y=1;n=0">
        <prompt>
            <value expr="coupon_number"
                   class="digits" /> is not a valid coupon number.
            Would you like to try again?
        </prompt>
        <filled>
            <if cond="try_again==true">
                <clear namelist="coupon_number try_again"/>
            <else/>
                <!-- if the user does not want to enter another coupon
                    number, just transition to the add x items to
                    shopping cart page. -->
            <goto next="PurchaseStuffField.vxml" />
        </filled>
    </field>

```

```

        </if>
    </filled>
</field>
</form>
</vxml>

<?xml version="1.0"?>
<!--PurchaseStuffField.vxml-->
<!--This document demonstrates the field element with built-in grammars -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sam's Publishing 2001"/>
    <meta name="description"
        content="Add j items to a shopping cart,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="collect_books">
        <field name="num_of_books" type="number">
            <prompt bargein="true" timeout="4s">
                How many books would you like to order?
            </prompt>
        </field>

        <field name="confirm" type="boolean?y=1;n=0">
            <prompt bargein="true" timeout="4s">
                You have ordered <value expr="num_of_books" />
                <!-- number of books -->

                <!--singular or plural of book-->
                <value expr="'book' + (num_of_books != 1 ? 's' : '')"/>

                Is this correct?
            </prompt>
        </field>
        <filled namelist="num_of_books confirm">
            <if cond="confirm == false">
                <clear namelist="num_of_books confirm" />
            <elseif cond="num_of_books == 0"/>
                <prompt>
                    You have chosen not to add any books to your
                    shopping cart.
                </prompt>
                <goto next="WelcomeAudio.vxml" />
            <else/>
                <assign name="application.total_books"
                    expr="application.total_books
                        + Number(num_of_books)"/>
                <assign name="application.num_current_order"
                    expr="Number(num_of_books)"/>
        
```

```

        <goto next="#output_total" fetchtimeout="3s"
            fetchaudio="fetchaudio.wav"/>
    </if>
</filled>
</form>
</vxml>

<?xml version="1.0"?>
<!--CalculateCostField.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly" />
    <meta name="copyright"
        content="Sams Publishing 2001" />
    <meta name="description"
        content="Calculate the cost of a book order,
        to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
    <meta name="robots" content="index,follow" />
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST" />
    <meta name="maintainer" content="johnoreilly@lucent.com" />
    <meta name="application" content="Book Store" />
    <var name="added_cost" />
    <var name="added_books" expr="application.num_current_order" />

    <form id="calculate_price">
        <subdialog name="discount"
            src=".//CalculateDiscount.vxml#calculate_discount">
            <param name="quantity" expr="application.total_books" />
            <param name="unit_price" expr="24.95" />
            <param name="isCoupon" expr="application.has_coupon" />
            <filled>
                <assign name="application.total_book_cost"
                    expr="Number(discount.price)" />
                <!-- Keep track of the cost of the books added -->
                <var name="old_cost"
                    expr="application.total_order_cost" />
                <assign name="application.total_order_cost"
                    expr="application.total_book_cost
                        + application.total_tshirt_cost" />
                <!-- Cost of the books added = new total - old total. -->
                <assign name="added_cost"
                    expr="application.total_order_cost - old_cost" />
                <goto next="#output_total" />
            </filled>
        </subdialog>
    </form>

    <form id="output_total">
        <block>
            <audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
            <audio src="added.wav" fetchtimeout="2s"> added </audio>

```

```
<value expr="String(Math.round(added_books))"/>
<prompt cond="added_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="added_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="to.wav" fetchtimeout="2s">to </audio>
<audio src="yourcart.wav" fetchtimeout="2s">
    your shopping cart.
</audio>
<audio src="thecost.wav" fetchtimeout="2s">
    The cost of the
</audio>
<value expr="String(Math.round(added_books))"/>
<audio src="added.wav" fetchtimeout="2s">added</audio>

<prompt cond="added_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="added_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="is.wav" fetchtimeout="2s">is </audio>
<value expr="added_cost" class="currency" />.
<audio src="cartcontains.wav" fetchtimeout="2s">
    Your shopping cart now contains
</audio>
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="application.total_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="and.wav" fetchtimeout="2s"> and </audio>
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">
    <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
</prompt>
<prompt cond="application.total_shirts != 1">
    <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
</prompt>

<audio src="totalcost.wav" fetchtimeout="2s">
    The total cost of the items in your shopping cart is
</audio>
```

```

<value expr="application.total_order_cost" class="currency" />
</block>
</form>
</vxml>
```

(1) BookStoreRoot.vxml 程序

程序 BookStoreRoot.vxml 增加 3 个变量，即 num_current_order、has_coupon 和 asked_about_coupon。Num_current_order 是用户添加的图书数量。它在程序 PurchaseStuffField.vxml 中被赋予一个值，并且在 CalculateCostField.vxml 中被引用。Has_coupon 是个布尔值，当用户拥有优惠券时，它的值是 true，否则它的值是 false。has_coupon 在程序 GetCouponInfo.vxml 中被赋予一个值，并在程序 CalculateCostField.vxml 中被引用。Asked_about_coupon 也是个布尔值类型的变量。如果用户以前已经被问到他是否拥有优惠券，那么 asked_about_coupon 的值是 false。Asked_about_coupon 变量用于确保用户不会每次试图将商品添加给购物车时都要被重复问到他是否拥有优惠券。Asked_about_coupon 变量也在 GetCouponInfo.vxml 程序中被赋予了一个值。

(2) GetCouponInfo.vxml 程序

在 GetCouponInfo.vxml 程序中，用户首次被问到 “Do you have a coupon?”（你有优惠券吗？）指令 <field name=“wether_coupon” type=“boolean? y=1; n=0”> 告诉平台上载内置的布尔值语法，如果用户回答 “yes”，或者按下 DTMF 键 1，便将 true 值赋予字段变量 whether_coupon。如果用户回答 “no”，或者按下 DTMF 键 0，平台便将 false 赋予 whether_coupon。

由于你的营销部认为经常访问你的站点的大多数用户将是程序员，因此你使用指令 boolean? y=1; n=0 作为 type 的值，将 DTMF 的 type 默认值 1=“yes” 和 2=“no”，改为程序员比较熟悉的 1=“yes” 和 0=“no”。当字段变量 whether_coupon 被赋予一个值时，filled 元素就将被执行。这里的 asked_about_coupon 的值被设置为 true。如果 whether_coupon 的值是 false，那么应用程序将转入 PurchaseStuffField.vxml，否则将访问下一个字段 coupon_number。

当 coupon_number 字段被访问时，便向用户说出 “Please enter your five digit coupon number。”（请输入你的 5 位数优惠券号）。coupon_number 字段的运行特性与 whether_coupon 字段相似，差别在于平台并不上载布尔语法，而是上载内置的数字语法。输入信息只限于用语音说出和 DTMF 按键输入数字 0-9。用户要么口述这些数字，要么用 DTMF 按键输入这些数字。VoiceXML 平台不允许用户在应答时混合使用这两种数字输入方法。digits? length=5 用于设定用户应答时必须输入 5 个数字才算有效。如果用户输入 55555，has_coupon 将被赋予值 true，然后应用程序便转入 PurchaseStuffField.vxml。如果用户输入的数字不是 55555，优惠券将被判为无效，然后访问 try_again 字段。

与 whether_coupon 字段一样，try_again 字段也使用内置的布尔语法，它采用新的 DTMF 值 1 和 0。当用户输入一个有效的应答时，try_again 字段被赋予 true 或 false 值，同时 try_again 字段的 filled 代码块将被执行。如果 try_again 的值是 true，字段变量 coupon_number 和 try_again 将被清除。清除这两个变量后，就可以再次访问 coupon_number 和 try_again 这两个字段。当字段变量清除后，

应用程序将再次转到 `coupon_number`。这个循环将反复运行下去，直到用户输入了有效的优惠券号码或者决定不再重试为止。如果用户决定不再重试，应用程序便转至 `PurchaseStuffField.vxml`，而 `has_coupon` 仍然被赋予 `false` 值。

`GetCouponInfo.vxml` 中的所有字段都给属性 `modal` 赋予默认值 `false`。让 `modal` 的值等于 `false` 后，字段“作用域”之外的语法就可以有效运行。你将在“link 元素”这一节的内容中看到，当你将应用程序作用域的语法添加给你的应用程序时，将 `false` 值赋予 `modal` 是非常重要的。如果有任何一个字段要求全局语法无效，那么该字段的 `modal` 属性只需要被赋予 `true` 值即可。

(3) PurchaseStuffField.vxml 程序

当 `PurchaseStuffField.vxml` 程序执行时，将要访问 `num_of_books` 字段，下面的指令

```
<prompt bargein="true" timeout="4s"> How many books would you like to
➥order?</prompt>
```

将导致向用户说出“`How many books would you like to order?`”（你想订购几本图书）这句话。

提示的 `bargein` 属性设置为默认值 `true`。如果 `bargein` 的值是 `true`，当输出仍然在说出或播放时，就可以接受用户的应答。应答既可以口述，也可以通过 DTMF 按键来发出。当 `bargein` 的值是 `true` 时，如果用户中断应答的话，输出就会立即停止，同时用户的应答将被处理。如果 `bargein` 的值是 `false`，用户的应答将被忽略，而输出则继续进行。当应用程序要求用户收听完整的输出，就像输出是个广告时，`bargein` 的值应该设置为 `false`。

`Prompt` 元素的 `timeout` 属性用于告诉平台，如果在输出完成后的 4 秒钟内没有检测到应答，那么平台应该产生一个 `noinput` 事件。关于事件的有关情况，将在本章后面部分中的“事件”一节中进一步介绍。

当用户用一个数字给第一个提示作出应答时，便访问 `confirm` 字段。在这个字段中，请用户输入的图书数量将被重复，并且请用户确认这个数量是否正确的。正如在 `GetCouponInfo.vxml` 示例程序中一样，用户可以用“`yes`”，“`no`”，`1` 或 `0` 来应答，对应的布尔值将被赋予 `confirm` 字段变量。

与 `GetCouponInfo.vxml` 程序中的 `filled` 元素不同，这些元素是 `field` 是元素的子元素，而在 `PurchaseStuffField.vxml` 程序中的 `filled` 元素则是 `form` 元素的子元素。在 `GetCouponInfo.vxml` 程序中，当 `filled` 元素的父变量 `field` 被赋予一个值时，`filled` 元素就被执行。而在 `PurchaseStuffField.vxml` 程序中，要等到 `filled` 元素的 `namelist` 中的各个变量全部被赋予相应的值之后，才执行 `filled` 元素。

如果用户选择不确认他的 `num_of_books` 变量，那么 `num_of_books` 和 `confirm` 字段变量均被清除，并且退出 `filled` 程序块，然后再次访问 `num_of_books` 字段。如果用户确认了他的 `num_of_books` 变量，而该变量的值是 `0`，便向用户读出一条消息，说明该用户选择不将任何图书添加给他的购物车，然后应用程序便转到 `welcome` 页。最后，如果 `num_of_books` 的值不是 `0`，并且用户已经确认了这个数量，那么 `num_of_books` 便添加给 `total_books`，`num_current_order` 被设置为 `num_of_books`，然后应用程序转到 `CalculateCostField.vxml`。

`num_of_books` 是 `type` 属性为 `number` 的一个字段的变量：

```
<field name="num_of_books" type="number" />
```

尽管字段的语法类型是内置的数字语法，但是赋予字段变量num_of_books的结果属于类型String。由于total_books和num_current_order这两个变量都必须属于类型number，因此在num_of_books的值被赋予total_books和num_current_order之前，它的类型将被转换成number：

```
<assign name="application.total_books"
       expr="application.total_books + Number(num_of_books)" />
<assign name="application.num_current_order" expr="Number(num_of_books)" />
```

(4) CalculateCostField.vxml 程序

对程序CalculateCostField.vxml进行的修改很少。首先，added_books被赋予的值是num_current_order，而不是硬编码值200。同样，discount子对话框的param元素中包含的变量isCoupon被赋予的值是has_coupon，而不是以前的硬编码值false。最后，对total_books进行更新的assign元素将被删除，因为该逻辑已经转至PurchaseStuffField.vxml程序。

9.2.2 option元素

option元素用于设定一系列的有效选项和对应的值，以便在选项被认可时将对应的值赋予字段变量。option元素配有哪些属性：

- dtmf——dtmf是与选项相关的DTMF序列。
- value——value是某个选项被选定时赋予字段变量的字符串。value的默认值是封装在option元素中的文本。

程序举例

现在你必须扩展你的在线图书商店应用程序，使之能够添加T恤衫，而不仅仅是图书。另外，你可以将优惠券限制为只能用于购买图书。程序清单9-7中的代码就可以用于执行这些操作。

程序清单9-7 BookStoreRoot.vxml, AddItem.vxml, Purchase_ StuffField.vxml 和CalculateCostOption.vxml

```
<?xml version="1.0"?>
<!--Root document-->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>

    <meta name="description"
          content="Root document for our book store,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
```

```

<var name="total_shirts" expr="0" />
<var name="num_current_order" expr="0" />
<var name="has_coupon" expr="false" />
<var name="asked_about_coupon" expr="false" />
<var name="unit_book_price" expr="24.95" />
<var name="unit_tshirt_price" expr="19.95" />
<var name="current_item" expr="'book'" />

</vxmle>

<?xml version="1.0"?>
<!--AddItem.vxml-->
<vxmle version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>

    <meta name="description"
        content="Select Item to add, to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com" />
    <meta name="application" content="Book Store" />

    <form id="pick_item">
        <field name="item">
            <prompt>
                <enumerate>
                    To add <value expr="_prompt"/> to your shopping cart,
                    say <value expr="_prompt"/> or press
                    <value expr="_dtmf"/>.
                </enumerate>
            </prompt>
            <option dtmf="1" value="book"> books </option>
            <option dtmf="2" value="shirt"> tee shirts </option>
            <filled>
                <assign name="application.current_item" expr="item" />
                <goto expr="(item=='book' &&
                    application.asked_about_coupon==false)?
                    'GetCouponInfo.vxml' :
                    'PurchaseStuffOption.vxml'"/>
            </filled>
        </field>
    </form>
</vxmle>

<?xml version="1.0"?>
<!--PurchaseStuffOption.vxml-->
<!--This document demonstrates the field element with built-in grammars -->
<vxmle version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>

```

```
<meta name="copyright" content="Sams Publishing 2001" />

<meta name="description"
      content="Add, j items to a shopping cart,
              to access call (xxx)-xxx-xxxx" />
<meta name="keywords" content="VoiceXML Sams Books" />
<meta name="maintainer" content="johnnyo@sams.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />

<form id="collect_items">
    <field name="num_of_items" type="number">
        <prompt bargein="true" timeout="4s">
            How many <value expr="application.current_item + 's'" />
            would you like to order?
        </prompt>
    </field>

    <field name="confirm" type="boolean?y=1;n=0">
        <prompt bargein="true" timeout="4s">
            You have ordered <value expr="num_of_items" />
            <!-- number of items -->
            <!-- singular or plural of the item being added -->
            <value expr="application.current_item +
                (num_of_items != 1 ? 's' : '')" />

            Is this correct?
        </prompt>
    </field>
    <filled namelist="num_of_items confirm">
        <if cond="confirm == false">
            <clear namelist="num_of_items confirm" />
        <elseif cond="num_of_items == 0"/>
            <prompt>
                You have chosen not to add any
                <value expr="application.current_item + 's'" />
                to your shopping cart.
            </prompt>
            <goto next="WelcomeAudio.vxml" />
        <else/>
            <if cond="application.current_item=='book'" >
                <assign name="application.total_books"
                       expr="application.total_books
                             + Number(num_of_items)" />
            <else/>
                <assign name="application.total_shirts"
                       expr="application.total_shirts
                             + Number(num_of_items)" />
            </if>
            <assign name="application.num_current_order"
                   expr="num_of_items" />
        <goto next="CalculateCostOption.vxml" fetchtimeout="3s" />
    </filled>
</form>
```

```
        fetchaudio="fetchaudio.wav" />
    </if>
</filled>
</form>
</vxml>

<?xml version="1.0"?>
<!--CalculateCostOption.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>

    <meta name="description"
          content="Calculate the cost of a book order,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>
    <var name="added_cost" />
    <var name="added_items" expr="application.num_current_order" />

    <form id="calculate_price">
        <subdialog name="discount"
                  src=".//CalculateDiscount.vxml#calculate_discount">
            <param name="quantity"
                  expr="application.current_item=='book' ?
                        application.total_books :
                        application.total_shirts" />
            <param name="unit_price"
                  expr="application.current_item=='book' ?
                        application.unit_book_price :
                        application.unit_tshirt_price" />
            <param name="isCoupon"
                  expr="application.current_item=='book' ?
                        application.has_coupon : false" />
        <filled>
            <if cond="application.current_item == 'book'">
                <assign name="application.total_book_cost"
                      expr="Number(discount.price)" />
            <else/>
                <assign name="application.total_tshirt_cost"
                      expr="Number(discount.price)" />
            </if>
            <!-- Keep track of the cost of the books added -->
            <var name="old_cost"
                  expr="application.total_order_cost" />
            <assign name="application.total_order_cost"
                  expr="Number(application.total_book_cost +
                            application.total_tshirt_cost)" />
        </filled>
    </subdialog>
</form>
```

```

        expr="application.total_book_cost
              + application.total_tshirt_cost" />
    <!-- Cost of the books added = new total - old total. -->
    <assign name="added_cost"
           expr="application.total_order_cost - old_cost" />
    <goto next="#output_total" />
  </filled>
</subdialog>
</form>

<form id="output_total">
  <block>
    <audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
    <audio src="added.wav" fetchtimeout="2s"> added </audio>
    <value expr="String(Math.round(added_items))"/>

    <!-- output book, books, tshirts, or tshirts -->
    <prompt cond="added_items == 1 &&
                  application.current_item == 'book'">
      <audio src="book.wav" fetchtimeout="2s">book</audio>
    </prompt>
    <prompt cond="added_items != 1 &&
                  application.current_item == 'book'">
      <audio src="books.wav" fetchtimeout="2s">books</audio>
    </prompt>
    <prompt cond="added_items == 1 &&
                  application.current_item == 'shirt'">
      <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
    </prompt>
    <prompt cond="added_items != 1 &&
                  application.current_item == 'shirt'">
      <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
    </prompt>

    <audio src="to.wav" fetchtimeout="2s">to </audio>
    <audio src="yourcart.wav" fetchtimeout="2s">
      your shopping cart.
    </audio>
    <audio src="thecost.wav" fetchtimeout="2s">
      The cost of the
    </audio>
    <value expr="String(Math.round(added_items))"/>
    <audio src="added.wav" fetchtimeout="2s">added</audio>

    <!-- output book, books, tshirts, or tshirts -->
    <prompt cond="added_items == 1 &&
                  application.current_item == 'book'">
      <audio src="book.wav" fetchtimeout="2s">book</audio>
    </prompt>
    <prompt cond="added_items != 1 &&
                  application.current_item == 'book'">
      <audio src="books.wav" fetchtimeout="2s">books</audio>
    </prompt>
  </block>
</form>

```

```

</prompt>
<prompt cond="added_items == 1 && application.current_item == 'shirt'>
    <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
</prompt>
<prompt cond="added_items != 1 && application.current_item == 'shirt'>
    <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
</prompt>

<audio src="is.wav" fetchtimeout="2s">is </audio>
<value expr="added_cost" class="currency" />.
<audio src="cartcontains.wav" fetchtimeout="2s">
    Your shopping cart now contains
</audio>
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">
    <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="application.total_books != 1">
    <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="and.wav" fetchtimeout="2s"> and </audio>
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">
    <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
</prompt>
<prompt cond="application.total_shirts != 1">
    <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
</prompt>

<audio src="totalcost.wav" fetchtimeout="2s">
    The total cost of the items in your shopping cart is
</audio>
<value expr="application.total_order_cost" class="currency" />
</block>
</form>
</vxml>

```

1. BookStoreRoot.vxml

BookStoreRoot.vxml程序增加了3个变量。即unit_book_price、unit_tshirt_price和current_item。unit_book_price是一本书的费用，unit_tshirt_price是一件T恤衫的费用。unit_book_price和unit_tshirt_price均在程序 CalculateCostOption.vxml 中被引用。Current_item既可以指图书，也可以指T恤衫，这要根据用户添加给他的购物车的是哪种商品而定。在AddItem.vxml程序中current_item被赋予一个值，并且将在PurchaseStuffField.vxml 和CalculateCostOption.vxml这两个程序中被引用。

2. AddItem.vxml

AddItem.vxml用于建立一个字段选项列表，以便确定用户想要添加给购物车的商品。Item字段使用enumerate元素来输出当前存在的商品项目和选择商品项目所需要的命令。enumerate元素将在下一节中介绍。下面的option元素：

```
<option dtmf="1" value="book"> books </option>
```

用于确保如果用户口述“books”或者按下可后电话键“1”，字段变量item将被赋予book值。同样，下面这个option元素：

```
<option dtmf="2" value="shirt">tee shirts </option>
```

用于确保如果用户口述“tee”，“shirt”或者“tee shirt”，或者按下电话键“2”，变量item将被赋予shirt值。如果option元素不包括value属性，item将被赋予option元素中封装的文本的值，即tee shirts。

你会奇怪地发现“tee”和“shirt”都是对上面讲的第二个option元素的有效应答。执行item字段会使VoiceXML平台建立一个包含books, shirts, tee和tee shirts的口语的语法和包含1和2的DTMF语法。用于建立平台语法的一个原则说，每个特定的单词或任何单词的组合都是有效的应答。若要接受tee shirts这个应答，请在tee和shirts之间加一个下划线：

```
<option dtmf="2" value="shirt">tee_shirts </option>
```

使用生成的语法有一个好处，那就是这样的语法可以跨平台运行的，因为平台能够用它支持的格式在内部创建语法。

与前面介绍的示例程序中的field元素包含的filled元素一样，当item被赋予一个值时，便访问filled元素，这里current_item被赋予值item。接着，下面的指令

```
<goto expr="(item=='book' && application.asked_about_coupon==false)? 'GetCouponInfo.vxml' : 'PurchaseStuffOption.vxml'"/>
```

只不过是下面这个指令的一种更加密紧凑的写法

```
<if cond=" item=='book' && application.asked_about_coupon==false">
    <goto next="GetCouponInfo.vxml" />
<else/>
    <goto next="PurchaseStuffOption.vxml" />
```

将被执行。如果选定的商品是图书，并且以前没有询问用户是否拥有优惠券，那么应用程序将转至GetCouponInfo.vxml。否则，应用程序将跳到PurchaseStuffOption.vxml。

3. PurchaseStuffOption.vxml

使用PurchaseStuffOption.vxml，就可以消除引用图书的所有硬编码输出，然后用输出的current_item的值来取代它们，窗体idcollect_books和字段名num_of_books被更加通用的名字collect_items和num_of_items取代。最后，如果输入一个添加给购物车的商品的有效数量，并且该数量得到用户的确认，那么当current_item是图书时，total_books就被更新，当current_item是T恤衫时，total_shirts被更新。

4. CalculateCostOption.vxml

CalculateCostOption.vxml能够用更加通用的变量名added_items取代变量added_books。discount子对话框中名字为quantity, unit_price和isCoupon的param元素全部被更新。根据

current_item的值，quantity可以被赋予total_books值，也可以被赋予total_shirts值。同样，unit_price可以被赋予unit_book_price值，也可以被赋予unit_shirt_price值。如果current_item是book，并且has_coupon的值是true，那么isCoupon变量将被赋予值true，否则，它被赋予值false。在output_total窗体中，在短语“added x book (s) to your shopping cart”和“the cost of the x added book (s)”对book (s)的引用将改为输出book (s)或者t-shirt (s)，这要根据current_item的值而定。

9.2.3 enumerate元素

enumerate元素用于按照option元素或菜单选项元素出现的顺序来输出字段的option元素或菜单的选项元素中定义的语法项。

_prompt是菜单选项的口语语法项，而_dtmf则是菜单选项的DTMF语法项。

在AddItem.vxml程序中，下面的指令

```
<enumerate>
    To add <value expr="_prompt"/>
    to your shopping cart, say <value expr="_prompt"/>
    or press <value expr="_dtmf"/>.
</enumerate>
```

会产生这样的输出：“To add books to your shopping cart say books or press 1。To add tee shirts to your shopping cart say tee shirts or press 2。”（如果要将图书添加给你的购物车，请说books或者按电话键1。如果要将T恤衫添加给你的购物车，请说Tshirt，或者按电话键2。）使用enumerate元素，就可以按照文本列出的顺序，为每个option元素重复输出enumerate封装的文本，每次都替换当前option元素的_prompt和_dtmf的值。在第一个迭代输出中，如果当前option元素是下面这个值时

```
<option dtmf="1" value="book"> books </option>
```

_prompt是books，_dtmf是1。在第二个迭代输出中，如果当前option选项是下面这个值时

```
<option dtmf="2" value="shirt">tee shirts </option>
```

_prompt是tee shirts，_dtmf是2。当enumerate元素用在menu元素之中，就像用于option元素中时，enumerate的运行特性与choice元素相同。enumerate元素使用时也可以不带任何封装的文件。在这种情况下，<enumerate/>相当于

```
<enumerate><value expr="_prompt"></enumerate>
```

9.2.4 menu元素

menu元素与form元素一样，是VoiceXML中的两种对话框之一。根据本书第1章的定义，VoiceXML对话框用于描述应用程序向用户说出的提示信息，也用于定义和接收用户的应答，并且用于描述程序控制的流程。菜单包含的prompt元素用于将应用程序的输出信息传递给用户，它也包含用于建立语法的choice元素，以便定义和接收用户输入的信息，此外它还包含描述程序控制的choice元素和事件。

menu元素配有下列属性：

- id——id用于标识菜单。当使用 goto、submit和 link等元素从另一个对话框导航到该菜单时，就可以使用赋予id的值。
- scope——scope是指在菜单的choice元素中定义的菜单语法的作用域。如果scope 的值是它的默认值dialog，那么菜单的语法只在对话框中才有效。如果scope 的值是document，那么菜单的语法将在整个VoiceXML 文档中有效，或者如果文档是个根文档，那么语法将在整个应用程序中有效。
- dtmf——dtmf的值可以是 true，也可以是false 。当dtmf的值是true时，封装在menu元素中的 choice元素如果没有配备已经定义的 dtmf属性，那么它将被自动赋予一个 DTMF 值。

程序举例

程序清单9-8中的示例程序将一个主菜单添加给了welcome页。通过主菜单，用户可以将商品项添加给购物车，可以从购物车上删除某个商品项，可以记录关于自动售书的书店或任何商品的说明，可以列出和获得他的购物车中商品的合计费用，也可以进行结账。

程序清单9-8 welcomeMenu.vxml

```
<?xml version="1.0"?>
<!--WelcomeMenu.vxml-->
<!--This document demonstrates the menu and enumerate commands -->
<vxml version="1.0" application="BookStoreRoot.vxml">
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" content="Sams Publishing 2001"/>

  <meta name="description"
        content="Welcome page for Sams online bookstore,
        to access call (xxx)-xxx-xxxx"/>

  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <form id="welcome">
    <block>
      <prompt>
        <audio src="welcome.wav" caching="safe" fetchtimeout="4s"
               fetchhint="prefetch">
        <pros vol="+20%">
          <div type="paragraph">
            <emp level="strong">Welcome</emp>
            to Sam's Publishing's online bookstore.
            Our store allows you to conveniently purchase
            Voice Application Development with Voice XML
            books and t-shirts. You can also visit us on
            the web at <break msec="100" />
        </pros>
      </audio>
    </block>
    <says sub="W W W dot voice genie dot com ">
      www.voicegenie.com
    </says>
  </form>
</vxml>
```

```

        </div>
        </pros>
        </audio>
        </prompt>
        <goto next="#what_next" />
    </block>
</form>

<menu id="what_next" scope="document" dtmf="true">
    <prompt>
        You are at the main menu.
        To add an item or items say add or press 1.
        To remove an item or items say remove or press 2.
        To record comments about the automated book store or any of our
        products say record comments or press 3.
        To list and get the total cost of the items in your shopping cart
        say list or press 4.
        To check out say check out or press 5.
    </prompt>
    <choice next="AddItem.vxml" fetchaudio="fetchaudio.wav">add</choice>
    <choice next="RemoveItem.vxml" fetchaudio="fetchaudio.wav">
        remove
    </choice>
    <choice next="GetComments.vxml" fetchaudio="fetchaudio.wav">
        record comments
    </choice>
    <choice next="ListItems.vxml" fetchaudio="fetchaudio.wav">
        list
    </choice>
    <choice next="Checkout.vxml" fetchaudio="fetchaudio.wav">
        check out
    </choice>
</menu>
</vxml>
```

WelcomeMenu.vxml程序中的菜单被赋予的id是 what_next。该id 被welcome 窗体中的 goto 元素引用。当what_next菜单被执行时，概述用户的选项和每个选项的对应语法的提示便输出给用户。如果某个选项的语法是匹配的，应用程序就转至该选项的next 属性指定的程序文档。例如，如果用户的应答是“add”或者按下电话键“1”，那么应用程序就转至AddItem.vxml页。

与字段的选项列表一样，菜单的语法也是平台生成的。平台将来自what_next 菜单的每个 choice元素的封装文本组合起来，然后生成了口语语法，它包含add、remove、record、comments、record comments、check、out和check out。与option 元素生成的语法一样，封装在choice元素中的每个特定单词或单词的组合都是有效的应答。这说明为什么 what_next的口语语法包含了 chech、out、record 和comments。若要从 what_next的语法中去掉 check和out，请在choice元素中的check 与out之间加上一个下划线：

```
<choice next="Checkout.vxml" fetchaudio="fetchaudio.wav">check_out</choice>
```

你也可以用同样的方法从what_next的语法中去掉 record 和comments。

由于what_next的dtmf属性的值被设置成true，因此平台生成了包含数字1至5的DTMF 语法。

当菜单的dtmf属性设置为true时，菜单中的每个 choice元素均被赋予一个从1开始的 DTMF值。

what_next的scope属性被赋予值 document。这使得 what_next的语法能够在 welcome窗体中生效。如果用户中断 welcome窗体中的提示，那么用户的应答将由 what_next菜单来处理。

该菜单也可以使用像AddItem.vxml中所用的那种字段选项列表来编写。如果语法匹配的结果只需要转到另一个对话框，字段变量设置后就不再被引用，对话框中只需要一个字段，并且不需要与字段元素相关的属性，那么便使用菜单而不使用选项列表。例如，如果在对话框中需要用modal属性来关闭所有其他的语法，那么将使用字段选项列表而不使用菜单。

与AddItem.vxml中用于字段选项列表的提示内容不同，用于what_next 的prompt 元素的内容并不包括enumerate元素。之所以不使用 enumerate元素，原因是用于每个选项的操作和命令没有标准的模式。

9.2.5 choice元素

choice元素用于设定口语语法和/或用作菜单语法的DTMF语法项。如果一个选项的语法项被识别，那么该choice元素的next、event或expr等属性将用于确定下一步将执行什么程序。只有next、expr和event之一能够拥有用于每个choice元素的值。包含在choice 元素中的语法的作用域总是对话框。

choice 元素配有下列属性：

- dtmf——dtmf是与选项相关的 DTMF序列。
- next——next用于从一个对话框转到另一个对话框。它用于设定要访问的下一个对话框。
如果被访问的对话框位于另一个文档中，对话框的名字必须使用下面的句法。

<URL of next document > # <dialog name >

URL可以是当前文档的绝对地址，也可以是相对地址。如果没有设定对话框的名字，便访问URL位置上的VoiceXML文档中的第一个对话框。如果访问的对话框位于同一个文档中，对话框名字的句法将是# < dialog name >。

- event——event用于设定choice元素的语法被识别的结果所产生的事件。
- expr——expr的作用与next 相似，差别在于expr 是个表达式，其计算的结果是个URL。
- caching——caching属性的值可以是两个值中的一个，这两个值是fast和safe。如是caching 的值设置为默认值 fast ，那么如果存在文档的高速缓存拷贝，并且没有过期，那么就使用 next或expr中设定的 URL位置上的该文档的高速缓存拷贝。如果caching设置为值safe，那么不管平台以前是否对文档进行了高速缓存，均从URL下载该文档。
- fetchaudio——fetchaudio 的值设置为声音文件的URL，以便一边下载文档，一边播放该文档。如果本属性没有设置，则不播放任何声音文件。
- fetchhint——fetchhint的值可以是 safe，也可以是 prefetch， prefetch是它的默认值。如果 fetchhint的值是 prefetch ，那么可以在实际需要下一个文档之前提取该文档。如果fetchhint 的值是safe ，则不能提前提取下一个文档。
- fetchtimeout——fetchtimeout 是在生成error.badfetch.event事件之前平台为检索文档而等待的时间间隔。

在WelcomeMenu.vxml程序的what_next 菜单中，与每个choice元素相关的 DTMF均由 what_next的dtmf属性来决定。在整个程序举例中，与 choice元素相关的 DTMF 值是它在选项列表中它的位置。但是你也可以显式确定与每个选项相关的DTMF，方法是为choice元素的 dtmf

属性设定一个值。下面这个示例程序展示了选项的dtmf属性：

```
<menu id="what_next" scope="document">
    <prompt>
        To add an item or items say add or press 1.
        To remove an item or items say remove or press 2.
        To record comments about the automated book store or any of our
        products say record comments or press 3.
        To list and get the total cost of the items in your shopping cart
        say list or press 4.
        To check out say check out or press 9.
    </prompt>
    <choice dtmf="1" next="AddItem.vxml" fetchaudio="fetchaudio.wav">
        add
    </choice>
    <choice dtmf="2" next="RemoveItem.vxml" fetchaudio="fetchaudio.wav">
        remove
    </choice>
    <choice dtmf="3" next="GetComments.vxml" fetchaudio="fetchaudio.wav">
        record comments
    </choice>
    <choice dtmf="4" next="ListItems.vxml" fetchaudio="fetchaudio.wav">
        list
    </choice>
    <choice dtmf="9" next="Checkout.vxml" fetchaudio="fetchaudio.wav">
        check out
    </choice>
</menu>
```

该程序中的what_next既可以接受DTMF输入，也可以接受口语输入。有的菜单也可能只接受DTMF输入。若要使菜单只接受DTMF输入，方法是删除每个choice元素的封装文本，然后通过menu或choice元素的dtmf属性，为每个选项设定一个DTMF值。下面的代码显示了一个只接受DTMF输入的菜单：

```
<menu id="what_next" scope="document" dtmf="true">
    <prompt>
        To add an item or items press 1.
        To remove an item or items press 2.
        To record comments about the automated book store or any of our
        products press 3.
        To list and get the total cost of the items in your shopping cart
        press 4.
        To check out press 5.
    </prompt>
    <choice next="AddItem.vxml" fetchaudio="fetchaudio.wav" />
    <choice next="RemoveItem.vxml" fetchaudio="fetchaudio.wav" />
    <choice next="GetComments.vxml" fetchaudio="fetchaudio.wav" />
    <choice next="ListItems.vxml" fetchaudio="fetchaudio.wav" />
    <choice next="Checkout.vxml" fetchaudio="fetchaudio.wav" />
</menu>
```

菜单也可以包含只接受口语输入的语法。若要建立一个只接受口语输入的菜单，请不要在menu元素或任何 choice元素中设定任何 dtmf 属性。下面的代码显示了一个只接受口语输入的菜单：

```
<menu id="what_next" scope="document">
  <prompt>
    To add an item or items say add.
    To remove an item or items say remove.
    To record comments about the automated book store or any of our
    products say record comments.
    To list and get the total cost of the items in your shopping cart
    say list.
    To check out say check out.
  </prompt>
  <choice next="AddItem.vxml" fetchaudio="fetchaudio.wav">add</choice>
  <choice next="RemoveItem.vxml" fetchaudio="fetchaudio.wav">
    remove
  </choice>
  <choice next="GetComments.vxml" fetchaudio="fetchaudio.wav">
    record comments
  </choice>
  <choice next="ListItems.vxml" fetchaudio="fetchaudio.wav">list</choice>
  <choice next="Checkout.vxml" fetchaudio="fetchaudio.wav">
    check out
  </choice>
</menu>
```

9.2.6 link元素

当link元素中的包含语法项被识别时，link元素便根据其属性next、expr或event，转至一个新的对话框，或者发出一个事件。对于每个link 元素来说，它只能拥有next、expr和event等属性之一的一个值。

link 元素可以是vxml元素和 form 元素的子元素，也可以是foum 窗体的子元素。link元素中包含的语法不能设定作用域。link元素的父元素负责确定 link 元素中包含的语法的作用域。表9-1显示了不同父元素的链接的语法的作用域。

表9-1 不同父元素的链接的语法的作用域

语法包含在作为下列元素的子元素的链接中	语法的作用域
字段项	字段项
窗体	对话框
Vxml (非根文档)	文档
Vxml (根文档)	应用程序

link元素配有哪些属性：

- next——next用于设定要访问的下一个对话框。如果被访问的对话框位于另一个文档中，那么该对话框的名字必须使用下面的句法

```
<URL of next document> # <dialog name>
```

该句法中的URL可以是到达当前文档的绝对地址，也可以是相对地址。如果没有设定对话框的名字，便访问URL位置上的VoiceXML文档中的第一个对话框。当被访问的对话框位于同一个文档中时，对话框的名字的句法将是#<dialog name>。

- event——event用于设定产生的一个事件，作为链接的元素的语法被识别的结果。
- expr——expr的作用与next相似，差别在于expr是个表达式，其计算的结果是个URL。
- caching——caching属性的值可以是两个值中的一个，这两个值是fast 和safe。如是caching的值设置为默认值fast，那么如果存在文档的高速缓存拷贝，并且没有过期，那么就使用next或expr中设定的URL位置上的该文档的高速缓存拷贝。如果caching设置为值safe，那么不管平台以前是否对文档进行了高速缓存，均从URL下载该文档。
- fetchaudio——fetchaudio的值设置为声音文件的URL，以便一边下载文档，一边播放该文档。如果本属性没有设置，则不播放任何声音文件。
- fetchhint——fetchhint的值可以是safe，也可以是prefetch，prefetch是它的默认值。如果fetchhint的值是prefetch，那么可以在实际需要下一个文档之前提取该文档。如果fetchhint的值是safe，则不能提前提取下一个文档。
- fetchtimeout——fetchtimeout是在生成error.badfetch.event事件之前平台为检索文档而等待的时间间隔。

程序举例

接着你可以添加各项功能，使用户只要说出“operator”或者按*0键，就可以在应用程序的任何部分上转到某个操作员，或者说“main menu”，“start”，“start over”或“home”，就可以从应用程序的任何位置上返回到主菜单。程序清单9-9显示了该代码。

程序清单9-9 BookStoreRoot.vxml 和WelcomeMenuLink..vxml

```
<?xml version="1.0"?>
<!--Root document -->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>
    <meta name="description"
          content="Root document for our book store,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <!-- variables -->
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
```

```

<var name="total_shirts" expr="0" />
<var name="num_current_order" expr="0" />
<var name="has_coupon" expr="false" />
<var name="asked_about_coupon" expr="false" />
<var name="unit_book_price" expr="24.95" />
<var name="unit_tshirt_price" expr="19.95" />
<var name="current_item" expr="book"/>

<!-- links -->
<link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
    <grammar type="application/x-jsgf"> operator </grammar>
    <dtmf type="application/x-jsgf"> "*" 0 </dtmf>
</link>

<link next="WelcomeMenuLink.vxml#what_next"
      fetchaudio="fetchaudio.wav">
    <grammar type="application/x-jsgf">
        main menu | start [over] | home
    </grammar>
</link>
</vxmld

<?xml version="1.0"?>
<!--WelcomeMenuLink.vxml-->
<!--This document demonstrates the link command -->
<vxmld version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
          content="Welcome page for Sams online bookstore,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="welcome">
        <block>
            <prompt>
                <audio src="welcome.wav" caching="safe" fetchtimeout="4s"
                       fetchhint="prefetch">
                    <pros vol="+20%">
                        <div type="paragraph">
                            <emp level="strong">Welcome</emp>
                            to Sams Publishing's online bookstore.
                            Our store allows you to conveniently purchase
                            Voice Application Development with Voice XML
                            books and t-shirts. You can also visit us on
                            the web at <break msec="100" />
            <sayas sub="W W W dot voice genie dot com ">
                www.voicegenie.com

```

```

        </sayas>
    </div>
</pros>
</audio>
To speak to an operator at any time, say operator or press star
zero. Another useful command is <emp> main menu </emp>.

</prompt>
<goto next="#what_next" />
</block>
</form>

<menu id="what_next" scope="document" dtmf="true">
<prompt>
    You are at the main menu.
    To add an item or items say add or press 1.
    To remove an item or items say remove or press 2.
    To record comments about the automated book store or any of our
    products say record comments or press 3.
    To list and get the total cost of the items in your shopping cart
    say list or press 4.
    To check out say check out or press 5.
</prompt>
<choice next="AddItem.vxml" fetchaudio="fetchaudio.wav">add</choice>
<choice next="RemoveItem.vxml" fetchaudio="fetchaudio.wav">
    remove
</choice>
<choice next="GetComments.vxml" fetchaudio="fetchaudio.wav">
    record comments
</choice>
<choice next="ListItems.vxml" fetchaudio="fetchaudio.wav">list</choice>
<choice next="Checkout.vxml" fetchaudio="fetchaudio.wav">
    check out
</choice>
</menu>
</vxml>

```

BookStoreRoot.vxml程序将两个link元素添加给了在线书店应用程序。当用户说出“operator”或按下*0键时，下面的link元素：

```
<link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
```

便被执行。当访问该链接时，应用程序立即转至CallOperator.vxml文档。CallOperator.vxml文档将在第10章的“transfer元素”一节中介绍。

当用户说出“main menu”、“stsrt”、“startover”或“home”时，下面的link元素

```
<link next="WelcomeMenuLink.vxml#what_next" fetchaudio="fetchaudio.wav">
```

被执行。当这个link元素被执行时，应用程序立即转至WelcomeMemuLink.vxml文档中的what_next菜单。在WelcomeMeimuLink.vxml文档的welcome窗体中，你会看到这样一个文本：“To speak to an operator at any time, say operator or press star zero.” Another useful command is

<emp> main menu</emp>（若要在任何时候与操作员说话，请说operator，或者按*0键。另一个可以使用的命令是<emp> main menu</emp>），添加这段文字的目的是告诉用户添加的link元素所具备的功能。

BookStoreRoot.vxml中的link元素包含的语法都配有相应的应用程序作用域，因为它们的父link元素是根文档中vxml元素的子元素。当访问field元素、subdialog元素或record元素时，若要关闭link元素中包含的语法，请将被访问的元素的modal属性设置为true。

在上面的示例程序中，link元素的next属性均被赋予相应的值。link元素产生事件的举例将在“事件”这一节中介绍。

9.2.7 grammar元素

grammar元素用于设定可能的口语输入，如果设定的输入被识别时，则返回一个字符串值。不同的VoiceXML平台支持不同的语法格式。使用grammar元素设定的语法既可以是从外部文件那里下载的、直接插入的用户定义的语法，也可以是内置的语法。直接插入的语法是在VoiceXML文档中设置的，封装在grammar元素之中。内置的语法类型是boolean、date、digits、currency、number和phone。

grammar元素配下列属性：

- src——src是包含一个口语语法的文件的URL。语法可以是内置的，也可以是用户定义的。内置的口语语法的格式是src=“builtin:grammar/<type>”，其中type是内置语法的类型。例如，内置口语语法类型boolean可以表示为src=“builtin:grammar/boolean”。
- scope——包含在field元素中的语法只能在该field元素内部有效，字段语法不能设定scope。包含在link元素中的语法也不能设定scope。链接语法的作用域始终是包含link元素的这个元素的作用域。

包含在form或menu元素中的语法既可以拥有document作用域，也可以拥有dialog作用域。如果作用域是默认值dialog，那么语法只在它的当前form或menu元素中有效。如果语法的作用域是document，那么该语法将在声明该语法的整个文档中和声明作为其根文档的语法的文档的任何文档中有效。语法的作用域也可以使用form或menu元素的scope属性来设定，而不使用语法元素的scope属性来设置。如果同时使用两种方法进行了设定，那么语法的作用域将是grammar元素中声明的scope值。

- type——type是指语法的格式。
- caching——caching属性的值可以是两个值中的一个，这两个值是fast和safe。如是caching的值设置为默认值fast，那么如果存在文档的高速缓存拷贝，并且它没有过期，那么就使用该语法的高速缓存拷贝。如果caching的值设置为safe，那么不管平台以前是否对语法进行了高速缓存，均从URL下载该语法。
- fetchhint——fetchhint的值可以是safe，也可以是prefetch，prefetch是它的默认值。如果fetchhint的值是prefetch，那么可以在实际需要下一个语法文件之前提取该文件。如果fetchhint的值是safe，则不能提前提取该语法。
- fetchtimeout——fetchtimeout是在生成error.badfetch.event事件之前平台为检索语法文件而

等待的时间间隔。

在BookstoreRoot.vxml中，下面的指令：

```
<grammar type="application/x-jsgf"> operator </grammar>
```

用于设定包含一个项目operator的口语语法。另外，下面这个指令：

```
<grammar type="application/x-jsgf"> main menu | start [over] | home </grammar>
```

用于设定包含项目main menu, start, start over和home的口语语法。上面这两种语法都是直插式语法，因为它们都是在VoiceXML文档内部设定的，并且封装在grammar元素内。

你可以不设定直插式语法，你也可以从外部文件那里下载语法。语法文件的URL可以用grammar元素的src属性来设定。例如：

```
<grammar type="application/x-jsgf"> main menu | start [over] | home </grammar>
```

也可以改写为

```
<grammar src="mainmenu.g" type="application/x-jsgf" />
```

其中mainmenu.g驻留的位置与BookStoreRoot.vxml相同，它包含：

```
#JSGF V1.0;
grammar mainmenu;
public <returncommands> = main menu | start [over] | home;
```

BookStoreRoot.vxml中的两个语法都是根据type属性值的规定，使用JSGF（Java语音语法格式）来设定的。在第二个指令中，符号|用于将每个语法项隔开。符号?用于指明它封装的单词是个选项单词。这就可以说明为什么语法中除了start over外，也包括了start。

在mainmenu.g中，#JSGF V1.0是语法文件的标题，它说明包含在语法文件中的语法是使用JSGF第一版设定的。Grammar mainmenu用于设定语法的名字。Mainmenu由语法规则returncommands组成。JSGF语法是一组规则。这些语法规则由可以口述的文字和/或者其他规则组成。若要了解外部语法的其他例子，请参见本章中的“initial元素”这一节的内容。若要了解JSGF的详细说明，请访问网址<http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/index.html>。

该应用程序运行时所在的平台必须将JSGF视为一种有效的语法格式，以便使应用程序能够正常地运行。在试图设定应用程序中的语法之前，或者在试图更改应用程序运行的平台之前，请查看平台的文档，了解它支持的语法类型和语法举例。

该程序例子中的grammar元素并没有设定scope属性。这些语法将自动拥有application作用域，因此不能设定它的作用域。其父元素是一个根文档的vxml元素的link元素中包含的所有语法都拥有application作用域。

9.2.8 dtmf元素

dtmf元素用于设定可能的DTMF按键，如果设定的输入被识别时，则返回一个字符串值。不同的VoiceXML平台支持不同的语法格式。使用dtmf元素设定的语法既可以是从外部文件那里下载的、直接插入的用户定义的语法，也可以是内置的语法。直接插入的语法是在VoiceXML文档中设置的，并且封装在dtmf元素之中。内置的语法类型是boolean、date、digits、currency、

number和phone。

dtmf元素拥有下列属性：

- src——src是包含一个DTMF语法文件的URL。语法可以是内置的，也可以是用户定义的。

内置的DTMF语法的格式是 src=“builtin: dtmf /<type>” ,其中 type 是内置语法的类型。例如，内置DTMF语法类型boolean 可以表示为src=“builtin: dtmf /boolean”。内置语法可以被取代。下面的代码将用值true=1和false=0来取代默认的DTMF值true = 1和false=2：

```
src="builtin:dtmf/boolean?y=1, n=0"
```

- scope——dtmf元素的scope属性的运行特性与grammar元素的 scope属性相同。由于 field元素中包含的 DTMF语法只能在该字段元素中有效，因此field元素的 DTMF语法不能设定它的作用域。link元素中包含的DTMF语法也不能设定作用域。链接的 DTMF 语法的作用域总是与包含link元素的这个元素的作用域相同。

包含在form或menu元素中的 DTMF 语法既可以拥有document作用域，也可以拥有 dialog作用域。如果作用域是默认值dialog，那么 DTMF语法只在它的当前form或 menu元素中有效。如果 DTMF 语法的作用域是document，那么 DTMF 语法将在声明使用该语法的整个文档和作为其根文档的任何文档中有效。DTMF语法的作用域也可以使用 form或 menu元素的 scope 属性来设定，而不使用dtmf元素的 scope属性来设置。如果同时使用 form 或menu元素的 scope 和dtmf 元素的scope属性进行了设置，那么 DTMF 语法的作用域将是dtmf元素中声明的 scope 值。

- type——type是指语法的格式。

• caching——caching 属性的值可以是两个值中的一个，这两个值是fast 和safe 。如是 caching的值设置为默认值 fast ，那么如果存在DTMF语法的高速缓存拷贝，并且它没有过期，那么就使用该DTMF语法的高速缓存拷贝。如果 caching的值设置为 safe ，那么不管平台以前是否对DTMF语法进行了高速缓存，均从URL下载该语法。

• fetchhint——fetchhint的值可以是 safe，也可以是 prefetch， prefetch是它的默认值。如果 fetchhint的值是 prefetch ，那么可以在实际需要DTMF语法文件之前预先提取该文件。如果fetchhint 的值是safe ，则不能预先提取该DTMF语法。

• fetchtimeout——fetchtimeout 是在生成error.badfetch.event事件之前平台为检索DTMF语法文件而等待的时间间隔。

在BookStoreRoot.vxml文档中，下面的指令：

```
<dtmf type="application/x-jsgf"> "*" 0 </dtmf>
```

用于设定包含一个项目 *0 的DTMF 语法。该DTMF 语法是个直插式语法，因为它是在VoiceXML文档中设定的，并且封装在 dtmf元素之中。你可以不将语法设置为直插式语法，你可以从某个外部文件那里下载该语法。在 DTMF语法中，语法文件的 URL 是由dtmf元素的 src属性设定的，例如：

```
<dtmf type="application/x-jsgf"> "*" 0 </dtmf>
```

也可以改为

```
<dtmf src="operatordtmf" type="application/x-jsgf" />
```

其中，operatordtmf.g与 BookStoreRoot.vxml 文档驻留在同一个位置，并且包含下面的代码：

```
#JSGF V1.0;
grammar operatordtmf;
public <operatordtmf> = "*" 0;
```

正如type属性的值所示，DTMF语法是使用Java语音语法格式（JSGF）来设置的。*和#键码封装在引号之中，数字0到9则没有引号。

在operatordtmf.g文档中，#JSGF V1.0是衣服文件的标题，它表示文件中包含的语法是使用JSGF第1版设置的。Grammar operatordtmf用于设定语法的名字。Operatordtmf包含了名字也是operatordtmf的语法规则。JSGF语法就是一组规则。DTMF语法规则由DTMF项和/或其他规则组成。关于JSGF的详细信息，请访问网址<http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/index.html>。

该应用程序运行时所在的voiceXML平台必须将 JSGF视为能使该应用程序运行的有效语法格式。在试图设置应用程序中的语法之前，或者试图更换应用程序运行时所在的平台之前，请参阅平台的文档资料，了解它支持的语法和语法举例。

该程序举例中的dtmf元素并没有设定 scope 属性。语法能够自动拥有application作用域，因此它不能设定作用域。Link元素中包含的所有语法，如果 link元素的父元素是某个根文档的vxml元素，那么它们将自动拥有 application作用域。

语法冲突

如果用户说出一个应答，或者输入了 DTMF，而结果包含了多个有效语法，这将会出现什么样的问题呢？VoiceXML提供了下列语法的首选次序：

- 1) 包含在当前字段中的语法和包含在当前字段中的链接中的语法。
- 2) 包含在当前窗体或菜单中的语法和包含在当前窗体中的链接中的语法。
- 3) 文档作用域包含在当前文档中的窗体或菜单中的语法，不过窗体和菜单不是当前的窗体或菜单。另外，包含在当前文档的 vxml 元素中的link元素中的语法。
- 4) 文档作用域包含在当前文档的根文档中的语法，包括包含在根文档的 vxml 元素中的link元素中的语法。

9.3 事件

VoiceXML提供了一个用于妥善处理各种事件的机制。事件既可以是预先定义的，比如用户没有对提示作出应答；也可以是应用程序定义的，比如用户试图订购一本书的某一部分。事件可以是正常事件，比如用户请求帮助；也可以是出错事件，比如应用程序请求访问一个预录制的声音文件，而这个声音文件是用系统不支持的格式录制的。

在VoiceXML的事件模式中，事件既可以由平台产生，也可以由应用程序产生。这些事件可以由应用程序抓取和处理，也可以由平台抓取和处理。下面各小节将详细介绍事件的类型，如何产生和发送事件，以及如何处理事件等问题。

9.3.1 事件的类型

上一节讲到，事件可以是预先定义的，也可以是应用程序定义的，可以是正常事件，也可

以是出错事件。正常的预定义事件有下列几种：

- cancel——用户要求取消当前的输出。
- telephone.disconnect.hangup——用户已经结束呼叫。
- exit——用户要求退出。
- help——用户要求帮助。
- noinput——平台希望得到用户的应答，但是在出现超时之前，没有得到用户的任何响应。
- nomatch——平台检测到一个应答，但是这个应答与任何有效的语法都不一致。

预定义的出错事件有下列几种：

- error.badfetch——提取请求访问的资源（如声音文件或VoiceXML文档）失败了。这种失败的原因包括资源不存在，设定的 URL不正确，文档中有句法错误，发生了网络故障，或者请求访问的文件的访问许可权设置不正确。
- error.semantic——执行文档时遇到了运行期错误。
- error.noauthorization——用户要求对平台执行一项他没有被授权运行的操作。如果用户在 transfer元素中设定一个长途电话号码，而平台强制要求所有的转接电话都是本地电话，那么这就是一个未经授权的请求的例子。
- error.unsupported.format——用户请求访问的资源所用的格式是平台所不支持的。这种情况的例子有：平台不支持语法所用的格式，平台不支持录制的声音文件的格式等。
- error.unsupported.element——平台遇到一个它不支持的有效元素。例如，如果平台不能实现transfer元素，那么该平台就发出一个 error.unsupported.transfer 事件。

应用程序定义的错误类型应该遵循预定义错误类型的命名约定。例如，任何错误类型都應該使用前缀error。

9.3.2 默认的catch元素

当产生一个事件但是没有事件处理程序来处理该事件，那将会出现什么情况呢？VoiceXML平台拥有默认的事件处理程序，它能处理这样的事件。表9-2 描述了负责处理各种事件的默认处理程序的运行特性。

表9-2 默认的事件处理程序

默认的事件处理程序	运行结果
Cancel	停止输出当前的提示
Error	输出特定平台的消息，然后结束应用程序的运行
Exit	结束应用程序的运行
Help	输出特定平台的消息，提示计数器的数字递增，然后执行 reprompt元素
Noinput	提示计数器的数字递增，然后执行reprompt元素
Nomatch	上簇特定平台的消息，提示计数器的数字递增，然后执行 reprompt元素
Telephone.disconnect	停止执行应用程序
所用其他事件	停止执行应用程序

由于你的应用程序中没有设定事件处理程序，因此在应用程序运行时生成的事件将由平台来处理。例如，如果，用户在访问你的welcome 程序页中的菜单时说出“tricycle”，平台就会说出一条消息，指明用户的选择无效，然后重新显示菜单中的提示包含的文字。

9.3.3 catch元素

catch 元素包含了由处理平台发出的和应用程序发出的事件的逻辑方法。catch 元素可以是 document、dialog或 form元素的子元素。catch 元素的作用域与它的父元素相同。catch 配有下列属性：

- event——event 属性是指抓取的一个或多个事件。event可以是一个特定事件，也可以是一个事件类别，比如 <catch event= “error” />。
- count——count是指包含 catch元素的 document、dialog或 form项有效时发生的事件次数。只有当 count 属性等于发生的事件次数，或者count小于发生的事件次数，但是大于同一个作用域的任何 catch元素的值时， catch元素才对一个事件进行处理。Count的默认值是1。
- cond——如果cond 设置为它的默认值true， catch 元素便对特定的事件进行处理。如果 cond的值是 false ， 默认平台的异常处理程序便负责事件的处理。

程序举例

在程序清单9-10的示例程序中，你添加了noinput， nomatch， help 和exit 等事件的处理功能。你还定义了一个事件，添加了一个事件处理程序，使用户可以重复要求显示某个提示。事件处理程序添加给了根文档BookStoreRoot.vxml，这样,这些处理程序就可以在整个文档中有效。

程序清单9-10 BookStoreRoot.vxml

```
<?xml version="1.0"?>
<!--Root document -->
<vxm1 version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>
    <meta name="description"
          content="Root document for our book store,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <!-- variables -->
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />
    <var name="num_current_order" expr="0" />
    <var name="has_coupon" expr="false" />
```

```
<var name="asked_about_coupon" expr="false" />
<var name="unit_book_price" expr="24.95" />
<var name="unit_tshirt_price" expr="19.95" />
<var name="current_item" expr="'book'"/>

<!-- links -->
<link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
    <grammar type="application/x-jsgf"> operator </grammar>
    <dtmf type="application/x-jsgf"> "*" 0 </dtmf>
</link>

<link next="WelcomeMenuLink.vxml#what_next"
      fetchaudio="fetchaudio.wav">
    <grammar type="application/x-jsgf">
        main menu | start [over] | home
    </grammar>
</link>

<link event="exit">
    <grammar type="application/x-jsgf"> [good] bye </grammar>
</link>

<link event="online_bookstore.repeat">
    <grammar type="application/x-jsgf"> repeat </grammar>
</link>

<!-- event handlers -->
<catch event="noinput" count="1">
    I'm sorry, I could not hear you. Let's try again.
    <reprompt/>
</catch>

<catch event="noinput" count="2">
    I'm sorry, I still could not hear you.
    I am transfering you to an operator right now.
    <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
</catch>

<catch event="nomatch" count="1">
    I'm sorry, I didn't understand that.
    Let's try again.
    <reprompt/>
</catch>

<catch event="nomatch" count="2">
    I'm sorry, I still didn't understand that.
    I am transfering you to an operator right now.
    <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
</catch>
<catch event="exit">
    Thank you for shopping at Sams Online Bookstore.
    Have a nice day!
</catch>
```

```

        <exit/>
    </catch>

    <catch event="help">
        To speak to an operator at any time, say operator or
        press star zero.
        To return to the main menu, say main menu.
        To listen to a prompt again say repeat.
        To end this call say good bye.
        <reprompt/>
    </catch>

    <catch event="online_bookstore.repeat">
        <reprompt/>
    </catch>
</vxml>
```

以前，当用户说出单词“exit”时，就会产生一个exit事件。默认的平台exit事件处理程序将通过执行指令<exit/>来处理该事件。下面的指令：

```

<link event="exit">
    <grammar type="application/x-jsgf"> [good] bye </grammar>
</link>
```

将在识别词组“good bye”或单词“bye”时也产生一个exit事件。下面的指令：

```

<catch event="exit">
    Thank you for shopping at Sams Online Bookstore. Have a nice day!
    <exit/>
</catch>
```

将通过说出句子“Thank you for shopping at Sams Online Bookstore. Have a nice day！”（谢谢你到萨姆在线书店购物。祝你愉快！）执行exit命令。

当应用程序期望用户作出应答，而实际上并未应答时，便产生一个noinput事件，同时当前对话框将它的noinput事件计数器的值递增1。在上面的示例程序中，下面的代码

```

<catch event="noinput" count="1">
    I'm sorry, I could not hear you. Let's try again.
    <reprompt/>
</catch>
<catch event="noinput" count="2">
    I'm sorry, I still could not hear you.
    I am transferring you to an operator right now.
    <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
</catch>
```

被添加，以便处理这个noinput事件。当在当前对话框中第一次产生一个noinput事件时，便执行event=“noinput”，count=“1”的catch元素。系统对用户说出“ I'm sorry I could not hear you. Let's try again ”（对不起，我无法听到你说的话。让我们再来一次）这句话。接着，执行reprompt元素，从而导致程序的控制返回给产生事件的对话框或窗体，然后根据提示计数器和

prompt元素的 **count**属性，选定要输出的正确提示。关于 **reprompt** 元素的情况，将在“**reprompt**元素”小节中介绍。

当一个对话框运行之后，它的 **noinput** 计数器便复置为零。但是，如果仍然在当前对话框中运行时又产生了另一个 **noinput** 事件，便执行 **event=“noinput”，count=“2”** 的 **catch**元素。该事件处理程序将通知说，她将被转给另一个操作员，而应用程序则转至 **CallOperator.vxml**。假设应用程序没有转至 **CallOperator.vxml**，而是返回到前一个对话框，并且产生了第三个 **noinput** 事件。这时便再次执行 **event=“noinput”，count=“2”** 的 **catch**元素，因为它拥有任何 **noinput** 事件处理程序的最高 **count** 值。

当应用程序无法使用户的应答与已经安装的语法相匹配时，便会产生一个 **nomatch** 事件。在上面的示例程序中，**nomatch** 事件处理程序的运行方式与 **noinput** 事件处理程序相似，差别在于 **nomatch** 事件处理程序发出的消息是“*I'm Sorry I didn't understand that.*”（对不起，我不懂），而 **noinput** 事件处理程序发出的消息是“*I'm sorry I could not hear you .*”（对不起，我无法听到你的应答。）

catch语句能够抓取多个事件。例如，你可以将 **nomatch** 和 **noinput** 事件处理程序组合起来，当两个事件都被抓取时，只向用户输出一个通用消息。这个代码将类似下面的样子：

```
<catch event="noinput nomatch" count="1">
    What was that? Let's try again.
    <reprompt/>
</catch>

<catch event="noinput nomatch" count="2">
    I am having trouble processing your request.
    I am transferring you to an operator right now.
    <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
</catch>
```

下面这个指令

```
<link event="online_bookstore.repeat">
    <grammar type="application/x-jsgf"> repeat </grammar>
</link>
```

将在用户说出“repeat”（重复）时产生应用程序定义的 **online_bookstore.repeat** 事件。下面的指令：

```
<catch event="online_bookstore.repeat">
    <reprompt/>
</catch>
```

通过调用 **reprompt** 元素来处理 **online_bookstore.repeat** 事件。最后，通过告诉用户他可以使用整个应用程序范围的命令，然后调用 **reprompt** 元素，从而对 **help** 事件进行处理。

事件处理中的冲突

如果有两个事件处理程序，它们处理的事件类型相同，事件计数器的 **count** 值相同，它们同时有效，并且发生了一个同时符合这两个处理程序处理条件的事件，那么就会发生“事件处理

冲突”。究竟选择哪个事件处理器来处理该事件，这要由作用域来决定。选择作用域的优先次序是form项, dialog, document, application, default。

在程序清单9-11的示例程序中，你首先添加了每个对话框特定的 help 事件处理器。这些事件处理器将输出特定对话框的信息，然后转至BookStoreRoot.vxml中的 help事件处理器，以便输出应用程序的常用命令。

程序清单9-11 BookStoreRoot.vxml 和WelcomeMenuEvent.vxml

```

<?xml version="1.0"?>
<!--Root document --&gt;
&lt;vxml version="1.0" &gt;
    &lt;meta name="author" content="John O'Reilly" /&gt;
    &lt;meta name="copyright"
          content="Sams Publishing 2001" /&gt;
    &lt;meta name="description"
          content="Root document for our book store, to access
    ↪call (xxx)-xxx-xxxx"/&gt;
    &lt;meta name="keywords" content="VoiceXML Sams Books" /&gt;
    &lt;meta name="maintainer" content="johnnyo@sams.net" /&gt;
    &lt;meta name="robots" content="index,follow" /&gt;
    &lt;meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST" /&gt;

    &lt;!-- variables --&gt;
    &lt;var name="total_order_cost" expr="0" /&gt;
    &lt;var name="total_book_cost" expr="0" /&gt;
    &lt;var name="total_tshirt_cost" expr="0" /&gt;
    &lt;var name="total_books" expr="0" /&gt;
    &lt;var name="total_shirts" expr="0" /&gt;
    &lt;var name="num_current_order" expr="0" /&gt;
    &lt;var name="has_coupon" expr="false" /&gt;
    &lt;var name="asked_about_coupon" expr="false" /&gt;
    &lt;var name="unit_book_price" expr="24.95" /&gt;
    &lt;var name="unit_tshirt_price" expr="19.95" /&gt;
    &lt;var name="current_item" expr="'book'" /&gt;
    &lt;var name="global_help" expr="true" /&gt;

    &lt;!-- links --&gt;
    &lt;link next="CallOperator.vxml" fetchaudio="fetchaudio.wav"&gt;
        &lt;grammar type="application/x-jsgf"&gt; operator &lt;/grammar&gt;
        &lt;dtmf type="application/x-jsgf"&gt; "*" 0 &lt;/dtmf&gt;
    &lt;/link&gt;

    &lt;link next="WelcomeMenuLink.vxml#what_next"
          fetchaudio="fetchaudio.wav"&gt;
        &lt;grammar type="application/x-jsgf"&gt;
            main menu | start [over] | home
        &lt;/grammar&gt;
    &lt;/link&gt;

    &lt;link event="exit"&gt;
</pre>

```

```

        <grammar type="application/x-jsgf"> [good] bye </grammar>
    </link>

    <link event="online_bookstore.repeat">
        <grammar type="application/x-jsgf"> repeat </grammar>
    </link>

    <!-- event handlers -->
    <catch event="noinput" count="1">
        I'm sorry, I could not hear you. Let's try again.
        <reprompt/>
    </catch>

    <catch event="noinput" count="2">
        I'm sorry, I still could not hear you.
        I am transfering you to an operator right now.
        <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
    </catch>

    <catch event="nomatch" count="1">
        I'm sorry, I didn't understand that.
        Let's try again.
        <reprompt/>
    </catch>
    <catch event="nomatch" count="2">
        I'm sorry, I still didn't understand that.
        I am transfering you to an operator right now.
        <goto next="CallOperator.vxml" fetchaudio="fetchaudio.wav" />
    </catch>

    <catch event="exit">
        Thank you for shopping at Sams Online Bookstore.
        Have a nice day!
        <exit/>
    </catch>

    <catch event="help">
        <assign name="global_help" expr="true" />
        To speak to an operator at any time, say operator or
        press star zero.
        To return to the main menu, say main menu.
        To listen to a prompt again say repeat.
        To end this call say good bye.
        <reprompt/>
    </catch>

    <catch event="online_bookstore.repeat">
        <reprompt/>
    </catch>
</vxml>

<?xml version="1.0"?>
<!--WelcomeMenuEvent.vxml-->

```

```

<!--This document demonstrates event handlers -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Welcome page for Sams online bookstore to access
    <call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="welcome">
        <block>
            <prompt>
                <audio src="welcome.wav" caching="safe" fetchtimeout="4s"
                    fetchhint="prefetch">
                    <pros vol="+20%">
                        <div type="paragraph">
                            <emp level="strong">Welcome</emp>
                            to Sams Publishing's online bookstore.
                            Our store allows you to conveniently purchase
                            Voice Application Development with Voice XML
                            books and t-shirts. You can also visit us on
                            the web at <break msecs="100" />
                            <sayas sub="W W W dot voice genie dot com ">
                                www.voicegenie.com
                            </sayas>
                        </div>
                    </pros>
                </audio>
                To speak to an operator at any time, say operator or press star
                zero. Other useful commands are main menu, repeat, and good
                bye.
            </prompt>
            <goto next="#what_next" />
        </block>
    </form>

    <menu id="what_next" scope="document" dtmf="true">
        <prompt>
            You are at the main menu.
            To add an item or items say add or press 1.
            To remove an item or items say remove or press 2.
            To record comments about the automated book store or any of our
            products say record comments or press 3.
            To list and get the total cost of the items in your shopping cart
            say list or press 4.
            To check out say check out or press 5.
        </prompt>
        <choice next="AddItemEvent.vxml" fetchaudio="fetchaudio.wav">
            add

```

```

</choice>
<choice next="RemoveItem.vxml" fetchaudio="fetchaudio.wav">
    remove
</choice>
<choice next="GetComments.vxml" fetchaudio="fetchaudio.wav">
    record comments
</choice>
<choice next="ListItems.vxml" fetchaudio="fetchaudio.wav">list</choice>
    <choice next="Checkout.vxml" fetchaudio="fetchaudio.wav">
        check out
    </choice>
    <catch event="help" cond="global_help">
        You are currently at the bookstore's main menu.
        <assign name="global_help" expr="false" />
        <throw event="help" />
    </catch>
</menu>
</vxml>

```

当用户在运行WelcomeMenuEvent.vxml程序的 what_next 菜单时说出 “help” 一词时，便选择what_next菜单中带有事件的help的 catch 语句，而不选择 BookStoreRoot.vxml 程序中带有事件 help 的catch语句。之所以选择 what_next 菜单中的事件处理程序，原因是它在选择作用域的优先次序中的位置高于BookStoreRoot.vxml程序中的事件处理程序（前者是对话框，后者是应用程序），同时也因为what_next中的处理程序的 cond 属性是true，因此该处理程序适合运行。

当what_next中的help事件处理程序运行时，它将输出what_next特定的消息：“You are currently at bookstore's main menu.”（现在你进入了书店的主菜单）。接着，该事件处理程序为应用程序的变量 Global_help赋予值 false，并且产生另一个help事件。由于 help事件处理程序的 cond 属性现在是false，它无法处理该事件，因此BookStoreRoot.vxml的 help事件处理程序开始执行。该处理程序输出应用程序的常用命令，并且将 global_help变量的值重新设置为 false，这样what_next的help事件处理程序就能够处理产生的下一个help事件，并且调用reprompt元素。reprompt元素将程序控制返回给what_next，并且再次播放提示。

9.3.4 throw元素

throw元素用于产生预定义的事件和应用程序定义的事件。它的event属性就是要产生的事件。

程序举例

在程序清单9-12的示例程序中，特定对话框的help 事件处理程序被添加给PurchaseStuffEvent.vxml程序。另外，由于商品的数量不能是分数或负数，因此增加了错误检查功能，以确保用户输入的数据是有效的。

程序清单9-12 PurchaseStuffEvent.vxml

```

<?xml version="1.0"?>
<!--PurchaseStuffEvent.vxml-->
<!--This document demonstrates event handling -->

```

```

<vxm version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Add j items to a shopping cart,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="collect_items">
        <catch event="error.invalid_quantity">
            I'm sorry, <value expr="num_of_items" /> is not a valid entry.
            Order quantities can not be fractions or negative.
            Let's try again.
            <clear namelist="num_of_items confirm" />
            <goto nextitem="num_of_items" />
        </catch>

        <field name="num_of_items" type="number">
            <prompt bargein="true" timeout="4s">
                How many <value expr="application.current_item + 's'" />
                would you like to order?
            </prompt>
            <catch event="help" cond="global_help">
                Say or enter using the number keys on your phone the quantity
                of <value expr="application.current_item + 's'" />

                you wish to add to your shopping cart.
                <assign name="global_help" expr="false" />
                <throw event="help" />
            </catch>
        </field>

        <field name="confirm" type="boolean?y=1;n=0">
            <prompt bargein="true" timeout="4s">
                You have ordered <value expr="num_of_items" />
                <!-- number of items -->
                <!-- singular or plural of the item being added -->
                <value expr="application.current_item +
                    (num_of_items != 1 ? 's' : '')"/>
                Is this correct?
            </prompt>
            <catch event="help" cond="global_help">
                If the quantity you entered is correct, say yes or press the
                one key on your phone. If you would like to reenter the
                quantity of <value expr="num_of_items" /> to add, say no or
                press the zero key on your phone.
            </catch>
        <assign name="global_help" expr="false" />
        <throw event="help" />
    </form>

```

```

        </catch>
    </field>
<filled namelist="num_of_items confirm">
    <!-- Only whole positive quantities will be be excepted -->
    <!-- Make sure the user entered an integer -->
    <var name="is_fraction"
        expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
    <var name="is_negative" expr="num_of_items < 0" />

    <if cond="confirm == false">
        <clear namelist="num_of_items confirm" />
    <elseif cond="num_of_items == 0"/>
        <prompt>
            You have chosen not to add any
            <value expr="application.current_item + 's'" />
            to your shopping cart.
        </prompt>
        <goto next="WelcomeAudio.vxml" />
    <elseif cond="is_fraction || is_negative" />
        <throw event="error.invalid_quantity" />
    <else/>
        <if cond="application.current_item=='book'" >
            <assign name="application.total_books"
                expr="application.total_books
                    + Number(num_of_items)" />
        <else/>
            <assign name="application.total_shirts"
                expr="application.total_shirts
                    + Number(num_of_items)" />
        </if>
        <assign name="application.num_current_order"
            expr="num_of_items" />
        <goto next="CalculateCostOption.vxml" fetchtimeout="3s"
            fetchaudio="fetchaudio.wav" />
    </if>
</filled>
</form>
</vxml>

```

在这个示例程序中，`help`事件处理程序被添加给`num_of_item`和`confirm`字段。当该字段的`help`事件处理程序执行时，首先输出特定字段的帮助信息。接着执行`BookStoreRoot.vxml`中的`help`事件处理程序，以便传递应用程序的常用帮助消息。

`number`的字段的有效输入项可以是任何正整数或负整数，也可以是小数和零。由于你的应用程序不允许将数量为分数或负数的图书添加给购物车，因此必须对`num_of_books`进行检查，以确保它不是负数或分数。当`filled`元素执行时，下面的指令

```
Math.ceil(num_of_books) != Math.floor(num_of_books)
```

将使用ECMAScript的`Math`函数`ceil`和`floor`来确定`num_of_books`究竟是分数还是整数。接着，`num_of_books`的值将与0进行比较，以确定它是否是个负数。

如果 num_of_books 的值是个负数或分数，便产生一个应用程序定义的出错事件 error.invalid_quantity :

```
<elseif cond="is_fraction || is_negative" />
<throw event="error.invalid_quantity" />
```

当发生 error.invalid_quantity 这个出错消息时，便访问 error.invalid_quantity 的事件处理程序：

```
<catch event="error.invalid_quantity">
    I'm sorry, <value expr="num_of_items" /> is not a valid entry.
    Order quantities can not be fractions or negative.
    Let's try again.
    <clear namelist="num_of_items confirm" />
</catch>
```

该事件处理程序首先给阅读者输出一个出错消息。接着 num_of_books 和 confirm 字段变量被清除，退出事件处理程序和 filled 元素，然后再次访问 num_of_books 字段。

9.3.5 error元素、help元素、noinput元素和nomatch元素

VoiceXML 提供了一种方法，以便非常方便地表示用于通常产生的事件 error、 help、 noinput 和 nomatch 的事件处理程序。使用速记表示法，事件 <catch event = “event”>...</catch> 可以写成 <event>...</event>，其中 event 可以是 error，也可以是 help、 noinput 或 nomatch。这些元素配有下列属性：

- count——count 是指包含 catch 元素的 document、 dialog 或 form 项有效时，某个事件发生的次数。catch 元素只处理这样的事件，即它的 count 的值等于或小于事件发生的次数，但是拥有相同作用域的任何 catch 元素的最大值。count 的默认值是 1。
- cond——如果 cond 的值设置为它的默认值 true， catch 元素将处理指定的事件。如果 cond 的值是 false，那么平台的默认异常处理程序将负责事件的处理。

程序举例

在程序清单 9-13 所示的 AddItemEvent.vxml 程序中，你将一个 help 事件处理程序添加给了 pick_item 窗体。

程序清单 9-13 AddItemEvent.vxml

```
<?xml version="1.0"?>
<!--AddItemEvent.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly" />
    <meta name="copyright"
        content="Sams Publishing 2001" />
    <meta name="description"
        content="Select Item to add, to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
    <meta name="robots" content="index,follow" />
    <meta http-equiv="Date"
```

```

        content="Thu, 7 Aug 2000 20:20:20 EST"/>
<meta name="maintainer" content="johnoreilly@lucent.com"/>
<meta name="application" content="Book Store"/>
<property name="caching" value="safe" />

<form id="pick_item">
    <field name="item">
        <prompt>
            <enumerate>
                To add <value expr="_prompt"/> to your shopping cart,
                say <value expr="_prompt"/> or press
                <value expr="_dtmf"/>.
            </enumerate>
        </prompt>
        <option dtmf="1" value="book"> books </option>
        <option dtmf="2" value="shirt"> tee shirts </option>
        <filled>
            <assign name="application.current_item" expr="item" />
            <goto expr="(item=='book' &&
                        application.asked_about_coupon==false)?
                        'GetCouponInfoEvent.vxml' :
                        'PurchaseStuffEvent.vxml'"/>
        </filled>
        <help cond="global_help">
            You are currently selecting an item to add to your shopping
            cart. Say the name of, or press the phone key corresponding
            to the item you would like to add.
        <assign name="global_help" expr="false" />
            <throw event="help" />
        </help>
    </field>
</form>
</vxml>
```

添加给pick_item窗体的 help事件处理程序的运行方式与 WelcomeMenuEvent.vxml 和 PurchaseStuffEvent.vxml示例程序中添加的help事件处理程序是相同的。但是，与这些示例程序中的事件处理程序不同的是，pick_item 窗体的事件处理程序是用速记表示法<help cond="global_help" ...</help> 来表示的。同样，你可以将BookStoreRoot.vxml 中的noinput 和 nomatch 事件处理程序表示为 <noinput count= "1" >...</noinput>和 <nomatch count= "1" >...</nomatch>。

9.3.6 reprompt元素

如果当前对话框或窗体中的提示的最大count属性小于或等于 prompt计数器值，那么 reprompt元素可用于使该提示得以播放。

程序举例

在程序清单9-4的示例程序中，你将help事件处理程序添加给GetCouponInfoEvent.vxml程序

中的 whether_coupon、coupon_number 和try_again 字段。此外，你修改了try_again中的提示，当第二次执行字段中的提示时，就可以向用户说出更简略的提示 “Would you like to try again ?” (你想再试一下吗？)。

程序清单9-14 GetCouponInfoEvent.vxml

```

<?xml version="1.0"?>
<!--GetCouponInfoEvent.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>
    <meta name="description"
        content="Get coupon from user if he or she has one, to access
    =>call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>
    <property name="caching" value="safe" />

    <form id="ask_for_coupon">
        <field name="whether_coupon" type="boolean?y=1;n=0">
            <prompt>
                Do you have a coupon?
            </prompt>
            <help cond="global_help">
                If you have a coupon, say yes or press the one key on your
                phone. If you do not have a coupon, say no or press the
                zero key on your phone.
            </help>
            <filled>
                <assign name="global_help" expr="false" />
                <throw event="help" />
            </filled>
            <assign name="application.asked_about_coupon" expr="true" />
            <if cond="whether_coupon==true" >
                <goto nextitem="coupon_number" />
            <else/>
                <!-- if the user does not have a coupon,
                    just transition to the add x items to the shopping
                    cart page. -->
                <goto next="PurchaseStuffEvent.vxml" />
            </if>
        </filled>
    </field>

    <field name="coupon_number" type="digits?length=5" >
        <prompt>

```

```

        Please enter your five digit coupon number?
    </prompt>
    <help cond="global_help">
        The coupon number should be located at the bottom of the
        email you received informing you of this special offer.
        Please either say or enter the number using the number keys
        on your phone.
    <assign name="global_help" expr="false" />
        <throw event="help" />
    </help>
    <filled>
        <!-- For now, all valid coupons will have the same number -->
        <if cond="coupon_number=='55555'">
            <assign name="application.has_coupon" expr="true" />
            <goto next="PurchaseStuffEvent.vxml" />
        <else/>
            <!-- if the user does not want to enter another coupon
                number, just transition to the add x items to
                the shopping cart page. -->
        <goto nextitem="try_again" />
        </if>
    </filled>
</field>

<field name="try_again" type="boolean?y=1;n=0">
    <prompt count="1">
        <value expr="coupon_number" class="digits" />
        is not a valid coupon number. Would you like to try again?
    </prompt>

    <prompt count="2">
        Would you like to try again?
    </prompt>

    <help cond="global_help">
        If you would like to enter your coupon number again,
        say yes or press the one key on your phone.
        If you would like to proceed with adding books to your
        shopping cart, say no or press the zero key on your phone.

        <assign name="global_help" expr="false" />
        <throw event="help" />
    </help>
    <filled>
        <if cond="try_again==true">
            <clear namelist="coupon_number try_again"/>
            <goto nextitem="coupon_number" />
        <else/>
            <!-- if the user does not want to enter another coupon
                number, just transition to the add x items to
                shopping cart page. -->
            <clear namelist="whether_coupon_coupon_number"

```

```

        try_again"/>
    <goto next="PurchaseStuffEvent.vxml" />
  </if>
</filled>
</field>
</form>
</vxml>

```

当第一次访问try_again字段时，它的提示计数器的值初始化为1。接着，下面的指令

```

<prompt count="1">
  <value expr="coupon_number" class="digits" /> is not a valid coupon number
  Would you like to try again?
</prompt>

```

被执行，并输出封装的文本，因为prompt计数器的值与提示的count属性1相一致。第一个提示执行后，下面的指令

```

<prompt count="2">
  Would you like to try again?
</prompt>

```

被执行。由于prompt计数器的值1与提示的count属性的值不一致，因此封装的文本不予输出。如果用户对第一个提示不作出应答，便产生一个noinput事件。BookStoreRoot.vxml程序中的noinput事件处理程序将调用reprompt元素：

```

<catch event="noinput" count="1">
  I'm sorry, I could not hear you. Let's try again.
  <reprompt/>
</catch>

```

reprompt元素将使try_again提示计数器的值递增1，然后再次执行try_again提示的语句。在这个迭代操作过程中，将输出count属性为2的提示，而count属性为1的提示则不输出。

9.3.7 initial元素

initial元素专门用于混合式初始窗体。混合式初始窗体包含作为窗体子元素的一个或多个初始元素和一个或多个语法。混合式初始窗体配有两个特性，一是窗体中包含的各个字段可以按任何顺序填写其信息，二是对语法的识别可以产生要填写的多个字段变量。

initial元素是个form项。用于向用户提示多个字段的信息。initial元素可以包含事件处理程序和提示，但是它可以不包含语法或filled元素。当执行initial程序块时，只有带有对话框、文档或应用程序的作用域的语法才有效。

initial元素配有下列属性：

- name——name是指与initial元素相关的一个变量。当访问一个变量时，如果name的值不是undefined，则不执行initial元素。
- expr——expr是name属性中声明的变量的初始值。expr不是必不可少的属性，如果它被省略了，那么变量的值就是undefined。
- cond——cond是个布尔值，用于确定initial中封装的指令是否在initial元素被访问时执行。

cond的默认值是true。

程序举例

在程序清单9-15的示例程序中，当用户在主菜单中选定Check Out 选项时，便提示用户输入订购的商品要发运到的国家。如果国家是美国，那么也可以在商品发运到的州进行货款托收。

程序清单9-15 CheckOut.vxml

```
<?xml version="1.0"?>
<!-CheckOut.vxml-->
<!-This document demonstrates the initial element -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Find out how long it will take to ship the items,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.met"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="get_location">
        <grammar src="statecountry.g" type="application/x-jsgf" />
        <initial name="start">
            <prompt>
                Please say the state, if in the United States, or the
                country you would like to ship your order to.
            </prompt>
            <help cond="global_help">
                You have chosen to check out.
                You are currently being asked for the shipping address for
                your order.
                <assign name="global_help" expr="false" />
                <throw event="help" />
            </help>
        </initial>

        <field name="state">
            <prompt>Which state would you like to send the order to? </prompt>
            <filled>
                <assign name="country" expr="'United States'" />
            </filled>
        </field>

        <field name="country">
            <filled>
                <if cond="country != 'United States'">
                    <assign name="state" expr="null"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

```

<field name="confirm" type="boolean?y=1;n=0">
    <prompt> You have requested to send your order to </prompt>
    <prompt cond="state != null">
        <value expr="state" />
    </prompt>
    <prompt>
        <value expr="country" />
        Is this correct?
    </prompt>
    <filled>
        <if cond="confirm == true">
            <prompt>
                Your order has been processed.
                Thank you for ordering from Sams Online Bookstore.
            </prompt>
        <else/>
            <clear namelist="start country state confirm" />
        </if>
    </filled>
</field>
</form>
</vxml>

#JSGF V1.0;
grammar stateandcountry;
public <statecountry> =
<state> {this.state=$} [<country> {this.country=$}] |
<country> {this.country=$};

<state> = Alabama | Alaska | Arizona | Arkansas | California | Colorado |
Connecticut | Delaware | Florida | Georgia | Hawaii | Idaho |
Illinois | Indiana | Iowa | Kansas | Kentucky | Louisiana | Maine |
Maryland | Massachusetts | Michigan | Minnesota | Mississippi |
Missouri | Montana | Nebraska | Nevada | New Hampshire | New Jersey |
New Mexico | New York | North Carolina | North Dakota | Ohio |
Oklahoma | Oregon | Pennsylvania | Rhode Island | South Carolina |
South Dakota | Tennessee | Texas | Utah | Vermont | Virginia |
Washington | West Virginia | Wisconsin | Wyoming;

<country> = United States | Canada | Mexico;

```

CheckOut.vxml 程序包含了一个混合式初始窗体的例子。语法statecountry.g有对话框的作用域。语法statecountry.g 包含3个语法规则，即state、country 和statecountry。state包含美国全部的50个州。country包括北美洲各国，即美国，加拿大和墨西哥。statecountry包括 state 和country的各项语法规则。它说明有效的应答将由state语法、先state语法后是country语法或country语法组成。statecountry语法规则中封装在{}中的文本说明已被识别的应答将被赋予封装的字段变量。例如，如果用户说出“Kansas”（堪萨斯州），那么{this.state= \$}就接受应答“Kansas”，并将它赋予state字段变量。

当访问get_location窗体时，执行的第一个元素是initial元素。initial元素将使提示“Please

say the state , if in the United States, or the country you would like to ship your order to.”（请说出你要将订购的商品发送到的州（如果在美国）或国家）得以播放。如果用户说出一个有效的州名，这个应答就被赋予字段变量state，然后访问state中字段中的filled元素。filled元素中的赋值语句将使字段变量country被赋予值United States。最后，cofirm字段被访问，同时要求用户确认他输入的信息。如果在initial提示之后用户说出一个有效的国家作为应答，该应答就被赋予字段变量country。接着，执行country字段中的filled元素。如果用户的应答是Canada或Mexico，那么State字段变量就被设置为null，然后访问confirm字段。如果用户的应答是United States，则退出country字段，然后访问state字段，因为它的字段变量的值仍然是undefined。当执行state字段时，便提示用户输入一个有效的州名。如果用户回答了一个有效的州名，便将用户的应答赋予state字段变量，然后访问confirm字段。

9.4 小结

在本章中，我们介绍了与输入/输出和事件处理有关的所有VoiceXML命令。输入/输出和事件处理命令构成了VoiceXML语言的主体。在本章中，我们给在线书店应用程序增加了与用户进行交互操作和恰当地处理各种事件的功能，使它的性能得到了增强。在下一章中，我们将要介绍剩余的一些VoiceXML元素，并且要完成在线书店应用程序中VoiceXML部分的编写。

第10章 VoiceXML的高级元素

本章将通过讲述剩余的元素record、transfer、script、property、object和submit，从而结束我们对VoiceXML语言的介绍，我们将要使用这些元素给在线书店应用程序添加功能。本章还要讲述每个元素的属性及其可能的取值。

10.1 record元素

record元素用于录制用户的口语输入。录制的声音文件可以通过 record元素的 name 属性中声明的变量来加以引用。record元素具有下列属性：

- name——name是引用用户录制的声音文件时使用的变量。如果访问 record 元素时， name 的值不是undefined，那么record元素将不执行。
- expr——expr是 name 属性中声明的变量的初始值。expr不是个必不可少的属性，如果它被省略，那么变量的值将是undefined。
- cond——cond 是个布尔值，用于确定访问record 元素时封装在该元素中的指令是否被执行。cond的默认值是true。
- modal——modal用于确定访问 record元素时口语语法和 DTMF 语法是否同时有效。如果 modal的值为默认值 true，那么其他语法就无效。无论modal的值是什么，大多数平台在录制声音文件时都不支持各种有效的语法。用户的所有应答都作为录音的一部分纳入声音文件。
- beep——如果beep的值是 true，那么当平台准备接受用户的输入时，就会向用户发出一声“嘀”。beep 的默认值是false。
- maxtime——maxtime 是指平台允许用户进行语音输入的最长时间。
- finalsilence——finalsilence 是指录音期间平台允许的静默时间。
- dtmfterm——如果dtmfterm的值是默认值 true，用户可以按下 DTMF键来结束录音。按键产生的声音将不是录音的一部分。
- type——type是录制的声音文件的格式。设定的 type必须是平台支持的声音文件格式。如果没有设定 type，将采用平台的默认格式来录制声音文件。

record 元素使用下列影子变量：

- name \$.duration——duration 是以毫秒为单位的录音长度。
- name \$.size——size是以字节为单位的录音文件的大小。
- name \$.termchar——termchar是为结束录音而输入的DTMF键。如果dtmfterm 的值是false，或者没有输入DTMF键，那么termchar就是 null。

程序举例

在程序清单10-1的示例程序中，在线书店应用程序添加的功能使用户能够对产品或应用程序

的说明进行录音。系统将要求用户在听到蜂音时录下他的说明，然后向用户播放此说明。

程序清单10-1 GetComments.vxml

```
<?xml version="1.0"?>
<!--GetComments.vxml-->
<!--This document demonstrates the record command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="This page collects comments for Sams online bookstore,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="record_comments">
        <record name="comments" modal="true" beep="true" maxtime="60s"
            finalsilence="2s" dtmfterm="true">
            <prompt>
                Please enter your comments at the beep.
            </prompt>
            <filled>
                <prompt>
                    The comments <value expr="comments" mode="recorded" />
                    have been saved.                                     </prompt>
                </filled>
            </record>
        </form>
    </vxml>
```

当GetComments.vxml 程序执行时，首先要求用户在听到蜂音时输入他的说明。beep属性设置为true，这样，当提示播放后，平台就产生一声“嘀”的蜂音。蜂音之后，用户开始录制他的说明。record的modal属性设置为true，这样，当用户录制他的说明时，所有语法均无效。平台将录制用户的说明，直到maxtime属性设定的60秒钟时间限制到期，或者根据finalsilence属性的规定，用户停止说话2秒钟为止。

当平台完成录音时，用户的口语说明将保存在一个声音文件中。变量comments用于引用该声音文件。当comments被赋予一个值后，便访问filled 程序块。在filled 程序块中，语句

```
<value expr="comments" mode="recorded" />
```

将使带有用户录制的说明的声音文件能够在文语转换（TTS）的输出 “The comments ”（说明）与 “have been saved” （已经保存）之间向用户回放。

10.2 transfer元素

transfer元素使得应用程序能够在用户与另一方之间启动一次电话呼叫。它具有下列属性：

- name——name是transfer变量的名字。当 transfer元素被访问时，如果 name 的值不是

`undefined`，那么 `transfer` 元素将不执行。当 `transfer` 操作完成时，`transfer` 变量将被赋予下列变量之一：

- `busy`——接收到一个忙音。
- `noanswer`——在 `connecttime` 属性规定的时间内没有收到回答。
- `network_busy`——电话网繁忙，无法接通电话。
- `near_end_disconnect`——当用户挂断电话时，电话已经接通，并且通话已经结束。
- `far_end_disconnect`——当呼叫方挂断电话时，电话已经接通，并且通话已经结束。
- `network_disconnect`——电话已经由网络接通并且又切断了。
- `expr`——`expr` 是 `name` 属性中声明的变量的初始值。`expr` 并不是必不可少的属性，如果它被省略，变量的值将是 `undefined`。
- `cond`——`cond` 是个布尔值，用于确定访问 `transfer` 元素时封装在该元素中的指令是否被执行。`cond` 的默认值是 `true`。
- `dest`——`dest` 是应用程序试图将电话转接到的一方的电话号码。
- `destexpr`——`destexpr` 是个表达式，用于计算应用程序试图将电话转接到的一方的电话号码。
- `bridge`——`bridge` 用于确定当电话的转接和后来的连接结束后呼叫方是否返回该应用程序。如果 `bridge` 的值是 `true`，当电话转接完成时，VoiceXML 平台将暂停应用程序的处理。当转换产生的电话呼叫结束时，VoiceXML 平台便恢复对应用程序的处理，同时，用户将恢复与应用程序之间的交互操作。如果 `bridge` 的值是 `false`，VoiceXML 平台将产生一个 `telephone.disconnect.transfer` 事件。这种类型的转接称为盲转，因为当电话转接完成时，用户的 VoiceXML 会话也就结束了。
- `connecttime`——`connecttime` 是平台在放弃转接电话的尝试并为 `transfer` 变量赋予 `noanswer` 值前为建立连接而等待的时间。
- `maxtime`——只有当 `bridge` 的值是 `true` 时，才能使用 `maxtime`。`maxtime` 是指用户返回他的 VoiceXML 会话前电话呼叫允许持续的时间。如果 `maxtime` 为默认值 0，则电话呼叫没有时间限制。

`transfer` 元素使用影子变量 `Name $.duration`，其中 `duration` 是以毫秒为单位的成功电话呼叫的时间长度。

程序举例

在前面的 `BookStoreRoot.vxml` 程序中已经引用过 `CallOperator.vxml` 程序：

```
<link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
<grammar type="application/x-jsgf"> operator </grammar>
<dtml src="operator.dtmf.g" type="application/x-jsgf"/>
</link>
```

在程序清单 10-2 中，实现了 `CallOperator.vxml` 程序。执行该程序时，用户将被转接到接线员。

程序清单10-2 CallOperator.vxml

```

<?xml version="1.0"?>
<!--CallOperator.vxml-->
<!--This document demonstrates the transfer command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="This page transfers the user to an operator,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form>
        <transfer name="call" dest="phone://9195558965" bridge="false"
            connecttimeout="8s" >
            <prompt>You are being transferred to an operator</prompt>
            <filled>
                <if cond="call == noanswer">
                    <prompt>
                        We're sorry, no operators are available at this time.
                        Please try calling again later. Good-bye.
                    </prompt>
                </if>
            </filled>
        </transfer>
    </form>
</vxml>

```

在这个示例程序中，平台说出一条消息，说明电话需立即转接到接线员。接着平台试图将用户连接到dest属性设定的虚构电话号码9195558965。如果电话在connecttimeout设定的8秒钟内成功接通，由于bridge属性的值等于false，那么用户在与接线员通话之后无法返回在线书店应用程序。如果8秒钟内没有接通，transfer的变量call将被赋予noanswer值，然后执行filled程序块。在filled程序块中，平台将说出一条消息，告诉用户转接失败，并请用户稍后再进行电话回叫。

10.3 script元素

使用script元素，ECMAScript代码既可以嵌入VoiceXML文档，也可以从某个外部信息源下载。Script元素具有下列属性：

- scr——scr是要执行的外部ECMAScript代码的URL。
- charset——charset是从外部下载代码时脚本程序的字符编码。
- caching——caching有两个可能的值，即fast和safe。如果caching设置为默认值fast，那么如果存在脚本的高速缓存拷贝并且没有过期，便使用该拷贝。如果caching的值设置为safe，那么不管平台是否在以前将脚本放入了高速缓存，都将从URL下载该脚本。

- **fetchhint**——**fetchhint**属性用于确定平台究竟是在实际需要之前就预先提取脚本（**fetchhint=prefetch**），还是在访问**script**元素时下载脚本（**fetchhhint=safe**）。
- **fetchtimeout**——**fetchtimeout**是指在产生**error.badfetch.event**事件前平台为检索脚本而等待的时间间隔。

程序举例

在线商店承诺，客户可以在两周内收到订购的商品，否则订购的商品将免费提供给用户。CheckOutScript.vxml程序（见程序清单10-3）中的**script**元素用于确定在线商店保证的商品到达日期，方法是取出现在日期，再加上14天。

程序清单10-3 CheckOutScript.vxml

```

<?xml version="1.0"?>
<!--CheckOutScript.vxml-->
<!--This document demonstrates the script element -->
<vxml version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Find out how long it will take to ship the items,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="get_location">
        <grammar src="statecountry.g" type="application/x-jsgf" />

        <initial name="start">
            <prompt>
                Please say the state, if in the United States, or the
                country you would like to ship your order to.
            </prompt>
            <help cond="global_help">
                You have chosen to check out.
                You are currently being asked for the shipping address for
                your order.
                <assign name="global_help" expr="false" />
                <throw event="help" />
            </help>
        </initial>

        <field name="state">
            <prompt>Which state would you like to send the order to? </prompt>
            <filled>
                <assign name="country" expr="'United States'" />
            </filled>
        </field>
    
```



```

        spoken_month = "August";
        break;
    case 8:
        spoken_month = "September";
        break;
    case 9:
        spoken_month = "October";
        break;
    case 10:
        spoken_month = "November";
        break;
    case 11:
        spoken_month = "December";
        break;
    }
</script>

<prompt>
    Your order has been processed. It will arrive by
    <value expr="spoken_month" />
    <value expr="String(Math.round(day))" />.
    Thank you for ordering from Sams Online Bookstore.
</prompt>
<else/>
    <clear namelist="start country state confirm" />
</if>
</filled>
</field>
</form>
</vxml>
```

在这个示例程序中，script元素声明了一个新的date对象，它以当前年、月、日为起始。接着，给date对象中month的day加上14，然后设置day和month变量，并执行switch语句。case语句将月份数字映射为月份的名字，再将该值赋予变量spoken_month。最后，货物到达的月份名字和日期在 script元素后面的 prompt 元素中向用户说出。

10.4 property元素

property元素用于控制语音识别、DTMF语法识别、提示、提取文档和资源，以及输入类型等的运行特性。你可以为应用程序、文档、对话框或窗体声明相应的属性。除了下面列出的标准属性外，平台还可以声明特定平台的属性。使用property 元素赋予的属性可以被使用带有更加特定作用域的property 元素赋予的属性取代，也可以被通过各个元素中的属性赋予的属性所取代。

property元素具有下列属性：

- confidencelevel——confidencelevel是0.0至1.0之间的一个数字。当平台识别的语法所具备的置信度低于confidencelevel时，便产生一个nomatch事件。如果confidencelevel的值是0.0，就代表返回的语法或数字的置信度很低。如果它的值是1.0，就表示返回的语法或数字的置

信度很高，本属性的默认值是0.5。

- **sensitivity**—sensitivity是0.0至1.0之间的一个数字。sensitivity用于确定用户的应答必须达到的音量。如果sensitivity的值是0，平台将不太容易让背景噪声触发产生一个能够被平台检测到的错误应答，而说话声音较小的人也难以让他们的口语应答被检测到。如果sensitivity 的值是1，那么说话声音较小的人的口语应答将比较容易被识别，而背景噪声也更加容易触发一个能被平台检测到的错误应答。sensitivity的默认值是0.5。
- **speedvsaccuracy**—speedvsaccuracy是个0.0至1.0之间的数字。speedvsaccuracy 用于设定平台究竟应该更重视迅速识别用户的应答，还是应该更加重视准确地识别用户的应答。当speedvsaccuracy的值为0时，平台最关注快速识别用户的应答。当speedvsaccuracy的值为1时，平台最关注准确识别用户的应答。它的默认值是0.5。
- **completetimeout**—completetimeout是单词被识别之后到平台返回匹配的语法之前的时间。例如，如果语法同时包含 book 和book store，那么completetimeout 是指平台识别book之后等待用户说出store的时间。本属性的默认值是根据不同平台的情况而定的。
- **incompletetimeout**—incompletetimeout是指单词被部分识别之后平台产生一个 nomatch事件之前的这段时间。例如，如果语法包含 North Carolina 这两个单词，而平台识别了用户应答的North，那么 incompletetimeout就是平台在产生 nomatch 事件之前等待用户说出“Carolina”一词的时间。本属性的默认值是根据不同平台的情况而定的。
- **interdigittimeout**—interdigittimeout 用于设定平台在返回已经输入的DTMF 之前为了用户输入更多的DTMF 而等待的时间。
- **termtimeout**—termtimeout是指最后一个 DTMF输入项被识别之后，平台返回已经输入的 DTMF之前的时间。例如，如果 DTMF 语法同时包含1和12，那么termtimeout就是平台识别了1之后和返回1之前等待用户输入2所花费的时间。本属性的默认值是根据不同平台的情况而定的。
- **termchar**—当用户输入与DTMF 匹配的termchar时，平台便立即返回以前收集的 DTMF 。本属性的默认值是#。
- **bargein**—如果bargein的值设置为默认值 true，用户可以用口语应答或DTMF 来中断输出。如果bargein的值是 false，用户不能中断输出，输出必须播放到它结束。
- **timeout**—提示播放后，平台产生noinput事件之前，它等待用户应答所花费的时间。timeout的默认值是根据不同平台的情况而定的。
- **caching**—caching 有两个可能的值，即fast和 safe。如果 caching的值设置为默认值 fast，那么如果存在文档或资源的高速缓存拷贝并且尚未过期，便使用该拷贝。如果caching的值设置为safe，那么不管平台以前是否对文档或资源进行了高速缓存，都将从 URL下载文档或资源。
- **audiofetchhint**—audiofetchhint用于确定平台是否应该在需要前就预先提取声音文件。如果audiofetchhint的值设置为safe，那么只有在需要时才下载声音文件。如果 audiofetchhint的值设置为默认值prefetch，那么平台就会在上载程序页时下载声音文件。如果 audiofetchhint的值是stream，那么平台不等声音文件下载完毕，便开始播放该文件。

- **documentfetchhint**—**documentfetchhint**的值可以是默认值safe，也可以是 prefetch。如果**documentfetchhint** 的值是prefetch，那么在实际需要之前就可以预先提取下一个文档。如果**documentfetchhint** 的值是safe，那么下一个文档就不能预先提取。
- **grammarfetchhint**—**grammarfetchhint**的值可以是 safe，也可以是默认值prefetch。如果**grammarfetchhint**的值是prefetch，那么可以在实际需要之前就预先提取语法文件。如果**grammarfetchhint**的值是 safe，那么语法文件不能预先提取。
- **objectfetchhint**—**objectfetchhint**的值可以是 safe，也可以是默认值 prefetch。如果**objectfetchhint** 的值是prefetch，那么在实际需要之前就可以预先提取对象代码和数据。如果**objectfetchhint**的值是safe，则不可以预先提取对象代码和数据。
- **scriptfetchhint**—**scriptfetchhint**的值可以是 safe，也可以是默认值 prefetch。如果**scriptfetchhint**的值是prefetch，那么在实际需要之前就可以预先提取外部脚本。如果**scriptfetchhint**的值是safe，则不可以预先提取外部脚本。
- **fetchaudio**—**fetchaudio**的值设置为下载文档时播放的声音文件的 URL。如果本属性没有设置，则不播放任何声音文件。
- **fetchtimeout**—**fetchtimeout**是指在平台产生**fetchtimeout**事件之前为检索文档或资源而等待的时间间隔。**Fetchtimeout**的默认值是根据不同平台的情况而定的。
- **inputmodes**—**inputmodes**属性用于确定用户如何输入他的应答的方式，如果 **inputmodes** 设置为它的默认值dmf voice，那么用户可以通过DTMF 或语音来输入他的信息。如果 **inputmodes**设置为 dmf，那么用户只能通过 DTMF 输入他的信息。如果**inputmodes** 设置为voice，则用户只能通过语音来输入他的信息。

程序举例

程序清单 10-4 中的示例程序将**property**元素添加给了你的在线书店应用程序。添加的**property**元素取代了各个元素的属性。带有名字**fetchaudio**的**property**元素被添加给**BookStoreRoot.vxml**程序，取代了该应用程序中的**choice**和**goto**元素的**fetchaudio**属性。同样，名字为**bargein**和**timeout**的**property**元素被添加给**PurchaseStuffProp.vxml**程序，以取代**PurchaseStuffProp.vxml**程序中的**Prompt**元素的**bargein**和**timeout**属性。此外，名字为**fetchtimeout**的**property**元素被添加给**CalculateCostProp.vxml**程序，以取代**audio**元素的**fetchtimeout**属性。

程序清单 10-4 BookStoreRoot.vxml

```
<?xml version="1.0"?>
<!--Root document -->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>
    <meta name="description"
          content="Root document for our book store,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
```

```

<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date"
      content="Thu, 7 Aug 2000 20:20:20 EST"/>
<!-- properties -->
<property name="fetchaudio" value="fetchaudio.wav"/>

<!-- variables -->
<var name="total_order_cost" expr="0" />
<var name="total_book_cost" expr="0" />
<var name="total_tshirt_cost" expr="0" />
<var name="total_books" expr="0" />
<var name="total_shirts" expr="0" />
<var name="num_current_order" expr="0" />
<var name="has_coupon" expr="false" />
<var name="asked_about_coupon" expr="false" />
<var name="unit_book_price" expr="24.95" />
<var name="unit_tshirt_price" expr="19.95" />
<var name="current_item" expr="'book'" />
<var name="global_help" expr="true" />

<!-- links -->
<link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
    <grammar type="application/x-jsgf"> operator </grammar>
    <!-- <dtmf type="application/x-jsgf"> "*" 0 </dtmf> -->
        <dtmf src="operatordtmf.g" type="application/x-jsgf"/>
</link>

<link next="WelcomeMenuLink.vxml#what_next"
      fetchaudio="fetchaudio.wav">
    <grammar src="mainmenu.g" type="application/x-jsgf" />
    <!-- <grammar type="application/x-jsgf">
        main menu | start [over] | home
    </grammar> -->
</link>

<link event="help">
    <grammar type="application/x-jsgf"> help </grammar>
</link>

<link event="exit">
    <grammar type="application/x-jsgf"> [good] bye </grammar>
</link>

<link event="online_bookstore.repeat">
    <grammar type="application/x-jsgf"> repeat </grammar>
</link>

<!-- event handlers -->
<catch event="noinput" count="1">
    I'm sorry I could not hear you. Let's try again.
    <reprompt/>

```

```
</catch>
<catch event="noinput" count="2">
    I'm sorry I still could not hear you.
    I am transferring you to an operator right now.
    <goto next="CallOperator.vxml" " />
</catch>

<catch event="nomatch" count="1">
    I'm sorry I didn't understand that.
    Let's try again.
    <reprompt/>
</catch>

<catch event="nomatch" count="2">
    I'm sorry I didn't understand that.
    I am transferring you to an operator right now.
    <goto next="CallOperator.vxml" " />
</catch>

<catch event="exit">
    Thank you for shopping at Sam's Online Bookstore.
    Have a nice day!
    <exit/>
</catch>

<catch event="help">
    <assign name="global_help" expr="true" />
    To speak to an operator at any time,
        say operator or press star zero.
    To return to the main menu.
    To listen to a prompt again say repeat.
    To end this call say good bye.
    <reprompt/>
</catch>

<catch event="error.online_bookstore.negative_shopping_cart_value">
    I'm sorry. Our system is experiencing technical difficulties.
    Please call back at a later time.
    <throw event="exit" />
</catch>

<catch event="online_bookstore.repeat">
    <reprompt/>
</catch>
</vxmle>
<?xml version="1.0"?>
<!--WelcomeMenuProp.vxml-->
<!--This document demonstrates the property element -->
<vxmle version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description">
```

```

        content="Welcome page for Sams online bookstore to access
→call (xxx)-xxx-xxxx"/>
<meta name="keywords" content="VoiceXML Sam Books"/>
<meta name="maintainer" content="johnnyo@sam.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
<form id="welcome">
    <block>
        <prompt>
            <audio src="welcome.wav" fetchtimeout="4s"
                fetchhint="prefetch">
                <pros vol="+20%">
                    <div type="paragraph">
                        <emp level="strong">Welcome</emp>
                        to Sams Publishing's online bookstore.
                        Our store allows you to conveniently purchase
                        Voice Application Development with Voice XML
                        books and t-shirts. You can also visit us on
                        the web at <break msecs="100" />

                        <sayas sub="W W W dot voice genie dot com ">
                            www.voicegenie.com
                        </sayas>
                    </div>
                </pros>
            </audio>
            To speak to an operator at any time,
            say operator or press star zero.
            Other usefull commands are main menu, repeat, and good bye.
        </prompt>
        <goto next="#what_next" />
    </block>
</form>

<menu id="what_next" scope="document" dtmf="true">
    <prompt>
        You are at the main menu.
        To add an item or items say add or press 1.
        To remove an item or items say remove or press 2.
        To record comments about the automated book store or
            any of our products say record comments or press 3.
        To list and get the total cost of the items in your shopping cart
            say list or press 4.
        To check out say check out or press 5.
    </prompt>
    <choice next="AddItemEvent.vxml">add</choice>
    <choice next="RemoveItem.vxml">remove</choice>
    <choice next="GetComments.vxml">record comments</choice>
    <choice next="ListItems.vxml">list</choice>
    <choice next="CheckOutScript.vxml">check out</choice>
    <catch event="help" cond="global_help">

```

```

        You are currently at the bookstore's main menu.
        <assign name="global_help" expr="false" />
        <throw event="help" />
    </catch>
</menu>
</vxm1>

<?xml version="1.0"?>
<!--PurchaseStuffProp.vxml-->
<!--This document demonstrates the property element -->
<vxm1 version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
          content="Add items to a shopping cart,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="collect_items">
        <property name="bargein" value="true"/>
        <property name="timeout" value="4s"/>
        <catch event="error.invalid_quantity">
            I'm sorry, <value expr="num_of_items" /> is not a valid entry.
            Order quantities can not be fractions or negative.
            Let's try again.
            <clear namelist="num_of_items_confirm" />
            <goto nextitem="num_of_items" />
        </catch>

        <field name="num_of_items" type="number">
            <prompt>
                How many <value expr="application.current_item + 's'" />
                would you like to order?
            </prompt>
            <catch event="help" cond="global_help">
                Say or enter using the number keys on your phone the quantity
                of <value expr="application.current_item + 's'" />
                you wish to add to your shopping cart.

                <assign name="global_help" expr="false" />
                <throw event="help" />
            </catch>
        </field>

        <field name="confirm" type="boolean?y=1;n=0">
            <prompt>
                You have ordered
                <value expr="num_of_items" /> <!-- number of items -->

```

```

<!-- singular or plural of the item being added -->
<value expr="application.current_item +
    (num_of_items != 1 ? 's' : '')"/>
Is this correct?
</prompt>
<catch event="help" cond="global_help">
    If the quantity you entered is correct, say yes or
    press the one key on your phone.
    If you would like to reenter the quantity of
    <value expr="application.current_item +
        (num_of_items != 1 ? 's' : '')"/> to add,
    say no or press the zero key on your phone.
    <assign name="global_help" expr="false" />
    <throw event="help" />
</catch>
</field>
<filled namelist="num_of_items confirm">
    <!-- Only whole positive quantities will be be excepted -->
    <!-- Make sure the user entered an integer -->
    <var name="is_fraction"
        expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
    <var name="is_negative" expr="num_of_items < 0" />

    <if cond="confirm == false">
        <clear namelist="num_of_items confirm" />
    <elseif cond="num_of_items == 0" />
        <prompt>
            You have chosen not to add any
            <value expr="application.current_item + 's'" />
            to your shopping cart.
        </prompt>
        <goto next="WelcomeMenuProp.vxml#what_next" />
    <elseif cond="is_fraction || is_negative" />
        <throw event="error.invalid_quantity" />
    <else/>
        <if cond="application.current_item=='book'" >
            <assign name="application.total_books"
                expr="application.total_books
                    + Number(num_of_items)" />
        <else/>
            <assign name="application.total_shirts"
                expr="application.total_shirts
                    + Number(num_of_items)" />
        </if>
        <assign name="application.num_current_order"
            expr="num_of_items" />
        <goto next="CalculateCostProp.vxml" fetchtimeout="3s" />
    </if>
</filled>
</form>
</vxml>
<?xml version="1.0"?>

```

```

<!--CalculateCostProp.vxml-->
<!--CalculateCostProp.vxml demonstrates the property element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sam's Publishing 2001"/>
    <meta name="description"
          content="Calculate the cost of an order,
          to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>
    <var name="added_cost" />
    <var name="added_items" expr="application.num_current_order" />

    <form id="calculate_price">
        <subdialog name="discount"
                   src=".//CalculateDiscount.vxml#calculate_discount">
            <param name="quantity"
                  expr="application.current_item=='book' ?
                        application.total_books :
                        application.total_shirts" />
            <param name="unit_price"
                  expr="application.current_item=='book' ?
                        application.unit_book_price :
                        application.unit_tshirt_price" />
            <param name="isCoupon"
                  expr="application.current_item=='book' ?
                        application.has_coupon : false" />
            <filled>
                <if cond="application.current_item == 'book'">
                    <assign name="application.total_book_cost"
                           expr="Number(discount.price)" />
                <else/>
                    <assign name="application.total_tshirt_cost"
                           expr="Number(discount.price)" />
                </if>
                <!-- Keep track of the cost of the books added -->
                <var name="old_cost"
                     expr="application.total_order_cost" />
                <assign name="application.total_order_cost"
                       expr="application.total_book_cost
                             + application.total_tshirt_cost" />
                <!-- Cost of the books added = new total - old total. -->
                <assign name="added_cost"
                       expr="application.total_order_cost - old_cost" />
                <goto next="#output_total" />
            </filled>

```

```
</subdialog>
</form>
<form id="output_total">
    <property name="fetchtimeout" value="2s"/>
    <block>
        <audio src="youhave.wav"> You have </audio>
        <audio src="added.wav"> added </audio>
        <value expr="String(Math.round(added_items))"/>

        <!-- output book, books, tshirts, or tshirts -->
        <prompt cond="added_items == 1 && application.current_item == 'book'">
            <audio src="book.wav">book</audio>
        </prompt>
        <prompt cond="added_items != 1 && application.current_item == 'book'">
            <audio src="books.wav">books</audio>
        </prompt>
        <prompt cond="added_items == 1 && application.current_item == 'shirt'">
            <audio src="tshirt.wav">t-shirt</audio>
        </prompt>
        <prompt cond="added_items != 1 && application.current_item == 'shirt'">
            <audio src="tshirts.wav">t-shirts</audio>
        </prompt>

        <audio src="to.wav">to </audio>
        <audio src="yourcart.wav">your shopping cart. </audio>
        <audio src="thecost.wav">The cost of the </audio>
        <value expr="String(Math.round(added_items))"/>
        <audio src="added.wav">added</audio>

        <!-- output book, books, tshirts, or tshirts -->
        <prompt cond="added_items == 1 && application.current_item == 'book'">
            <audio src="book.wav">book</audio>
        </prompt>
        <prompt cond="added_items != 1 && application.current_item == 'book'">
            <audio src="books.wav">books</audio>
        </prompt>
        <prompt cond="added_items == 1 && application.current_item == 'shirt'">
            <audio src="tshirt.wav">t-shirt</audio>
        </prompt>
        <prompt cond="added_items != 1 && application.current_item == 'shirt'">
            <audio src="tshirts.wav">t-shirts</audio>
        </prompt>

        <audio src="is.wav">is </audio>
```

```

<value expr="added_cost" class="currency" />
<audio src="cartcontains.wav">
    Your shopping cart now contains
</audio>
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">
    <audio src="book.wav">book</audio>
</prompt>
<prompt cond="application.total_books != 1">
    <audio src="books.wav">books</audio>
</prompt>

<audio src="and.wav"> and </audio>
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">
    <audio src="tshirt.wav">t-shirt</audio>
</prompt>
<prompt cond="application.total_shirts != 1">
    <audio src="tshirts.wav">t-shirts</audio>
</prompt>

<audio src="totalcost.wav">
    The total cost of the items in your shopping cart is
</audio>
<value expr="application.total_order_cost" class="currency" />

<goto next="WelcomeMenuProp.vxml#what_next" fetchtimeout="4s" />
</block>
</form>
</vxml>

```

在这个示例程序中，fetchaudio属性添加给了BookStoreRoot.vxml程序：

```
<property name="fetchaudio" value="fetchaudio.wav" />
```

结果，当在本应用程序中下载一个文档时，fetchaudio.wav文件就被播放。由于fetchaudio属性拥有应用程序作用域，因此在该应用程序中的 choice 和 goto 元素中表示的fetchaudio属性将没有必要再使用。例如，fetchaudio属性将从WelcomeMenuProp.vxml 程序的what_next菜单中的 choice 元素：

```
<choice next="AddItemEvent.vxml">add</choice>
```

和PurchaseStuffProp.vxml 程序的filled程序块中的 goto 元素中被删除。

```
<goto next="CalculateCostProp.vxml" fetchtimeout="3s" />
```

另外，在PurchaseStuffProp.vxml程序中，bargein 和 timeout 这两个属性被赋予下面的值：

```
<property name="bargein" value="true" />
<property name="timeout" value="4s" />
```

由于这些property元素都是collect_item窗体的子元素，因此它们的作用域是对话框。通过

property元素来定义这些属性，你就可以从该对话框中的各个prompt元素中删除有关的属性。

同样，在CalculateCostProp.vxml程序的output_total窗体中，定义了fetchtimeout属性：

```
<property name="fetchtimeout" value="2s" />
```

该property元素取代了各个audio元素的fetchtimeout属性。但是下面的goto元素

```
<goto next="WelcomeMenuProp.vxml#what_next" fetchtimeout="4s" />
```

则保留了它的fetchtimeout属性，并且取代了property元素中声明的fetchtimeout属性。结果WelcomeMenuProp.vxml文件的提取超时是4秒，而不是2秒。

10.5 object元素

使用object元素，VoiceXML开发人员就可以访问不是VoiceXML提供的特定平台的功能。由于有些对象是平台特定的对象，因此使用一个对象就会导致某个应用程序与特定VoiceXML平台相关联。如果平台不知道被引用的对象，便产生一个error.unsupported.object事件。大多数平台出于安全方面的考虑，都不允许使用开发人员定义的对象。

Object元素配有了下列属性：

- name——name是引用返回的对象时使用的变量。当访问一个object元素时，如果name的值不是undefined，那么将不执行object元素。
- expr——expr是name属性中声明的变量的初始值。Expr不是必不可少的属性，如果它被省略了，那么变量的值就是undefined。
- cond——cond是个布尔值，用于确定访问元素时是否执行封装在object元素中的指令。cond的默认值是true。
- classid——classid是对象所在设备的URL。
- codebase——codebase是确定classid、data和archive等元素中设定的所有相对URL时使用的基本URL。它的默认值是当前文档的基本URL。
- codetype——codetype是对象的内容类型。它的默认值是type属性中设定的内容类型。
- data——data是对象使用的数据的URL。例如，data可以是一个对象为了向以后提示某种信息而使用的声音文件的目录的URL。
- type——type是对象的数据的内容类型。
- archive——archive是个用空格分开的URL列表。URL包含了对象所用的资源档案。
- caching——caching有两个可能的值，即fast和safe。如果caching的值设置为默认值fast，那么如果存在对象的高速缓存拷贝并且尚未过期，便使用该拷贝。如果caching的值设置为safe，那么不管平台以前是否对该对象进行了高速缓存，都将从URL下载该对象。
- fetchaudio——fetchaudio的值设置为对象下载时播放的声音文件的URL。如果本属性没有设置，则不播放任何的声音文件。
- fetchhint——fetchhint用于确定在实际需要对象之前平台是预先提取该对象(fetchhint=prefetch)，还是只有在访问object元素时才下载该对象(fetchhint=safe)。
- fetchtimeout——fetchtimeout是平台在产生error.badfetch.event事件前为检索对象而等待的时间间隔。

程序举例

程序清单10-5中的示例程序引用了一个假设的对象，它使用的参数有信用卡公司、信用卡号码和到期日期等，并且返回交易是否被批准的信息。

程序清单10-5 ValidateCreditCard.vxml

```
<?xml version="1.0"?>
<!--ValidateCreditCard.vxml-->
<!--This example demonstrates a hypothetical credit card validator-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>
    <meta name="description"
        content="Validate a credit card, to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>
    <form>
        <object name="card"
            classid="java:voicexml.objects.ValidateCreditCard.class">
            <prompt>
                One moment while we validate your credit card.
            </prompt>
            <param name="amount" expr="application.total_order_cost"/>
            <param name="company" expr="credit_card_company"/>
            <param name="card_number" expr="credit_card_number"/>
            <param name="expiration_month" expr="exp_month"/>
            <param name="expiration_year" expr="exp_year"/>
            <filled>
                <prompt>
                    Your credit card has been
                    <value expr="card.status"/> <!--approved or denied-->
                    Please hold while we transfer you to the operator.
                </prompt>
                <goto next="CallOperator.vxml" />
            </filled>
        </object>
    </form>
</vxml>
```

在上面这个示例程序中，平台首先说“One moment while we validate your credit card”（请稍等一会儿，我们要检验你的信用卡），然后执行平台上的./voicexml/object 目录中的java类 ValidateCreditCard.class。这个URL是在classid属性中设置的。

平台使用param元素将参数 amount、card-number、expiration_month 和 expiration_year 传递给

ValidateCreditCard。ValidateCreditCard使用这些信息来检验交易的情况，检验的方式与你通过因特网购买商品时遇到的因特网付款方式相同。接着 ValidateCreditCard 返回一个ECMAScript 对象，应用程序将该对象赋予变量 card。card是在object元素的 name 属性中设定的。最后，执行对象中的filled 元素。在filled 程序块中，使用card对象的 status变量告诉用户，交易究竟是批准了还是被拒绝了，接着文档转至 CallOperator.vxml。

注意 得到最广泛支持的 VoiceXML 对象是SpeechObjects，这是由Nuance开发的一组预先定义并且可以重复使用的对话框。你可以通过 object元素访问 SpeechObject。SpeechObjects可用于语音身份识别，提示搜集用户的信用卡信息或电话号码等。关于 SpeechObjects的详细说明，请访问网址 www.v-commerce.com。

10.6 submit元素

submit元素用于从一个 VoiceXML文档转到服务器端的脚本文件，比如用Perl、CGI、JSP或ASP编写的文件。Submit元素可以用来取代 goto元素，以便将变量从当前VoiceXML文档传递给目的文件。

Submit元素配有下列属性：

- **next**——next是下一个要执行的脚本文件的 URL。URL可以是当前文档的绝对地址，也可以是当前文档的相对地址。如果 next 设定了一个值，那么就不能设定expr的值。
- **expr**——expr的作用与next 相同，差别在于expr是个表达式，其结果是计算得出的URL。如果有expr的值，就不能设定 next的值。
- **method**——method是指request（请求访问数据）的方法。它可以是默认值get也可以是post。
- **namelist**——namelist是提交下一个文件的变量的列表。如果namelist省略了，那么作用域内的所有字段项变量均被提交。
- **enctype**——enctype是提交的文档的 MIME 编码类型。它的默认值是application/x-www-form-urlencoded。
- **caching**——caching 有两个可能的值，即fast和 safe。如果 caching的值设置为默认值 fast，那么如果存在文档的高速缓存拷贝并且尚未过期，便使用该拷贝。如果caching的值设置为 safe，那么不管平台以前是否对文档进行了高速缓存，都将从 URL下载文档。
- **fetchtimeout**——fetchtimeout是产生 error.badfetch.event事件前平台为检索文档而等待的时间间隔。
- **fetchhint**——fetchhint的值可以是 safe，也可以是 prefetch， prefetch是它的默认值。如果 fetchhint的值是 prefetch，那么可以在实际需要下一个文档之前预先提取该文档。如果 fetchhint 的值是safe，则不能预先提取下一个文档。
- **fetchaudio**——fetchaudio的值设置为文档下载时播放的声音文件的 URL。如果本属性没有设置，则不播放任何的声音文件。

程序举例

程序清单 10-6 中的示例程序使用submit元素将添加给订单的商品数量传递给 Calculate

CostAdd小服务程序。

程序清单10-6 PurchaseStuffSubmit.vxml

```

<?xml version="1.0"?>
<!--PurchaseStuffSubmit.vxml-->
<!--This document demonstrates the submit element -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Add j items to a shopping cart,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

    <form id="collect_items">
        <property name="bargin" value="true"/>
        <property name="timeout" value="4s"/>
        <catch event="error.invalid_quantity">
            I'm sorry, <value expr="num_of_items" /> is not a valid entry.
            Order quantities can not be fractions or negative.
            Let's try again.
            <clear namelist="num_of_items confirm" />
            <goto nextitem="num_of_items" />
        </catch>

        <field name="num_of_items" type="number">
            <prompt>
                How many <value expr="application.current_item + 's'" />
                would you like to order?
            </prompt>
            <catch event="help" cond="global_help">
                Say or enter using the number keys on your phone the quantity
                of <value expr="application.current_item + 's'" />
                you wish to add to your shopping cart.
                <assign name="global_help" expr="false" />
                <throw event="help" />
            </catch>
        </field>

        <field name="confirm" type="boolean?y=1;n=0">
            <prompt>
                You have ordered
                <value expr="num_of_items" /> <!-- number of items -->
                <!-- singular or plural of the item being added -->
                <value expr="application.current_item +
                    (num_of_items != 1 ? 's' : '')"/>
            </prompt>
            Is this correct?
        </field>
    </form>
</vxml>

```

```

</prompt>
<catch event="help" cond="global_help">
    If the quantity you entered is correct, say yes or press the
    one key on your phone. If you would like to reenter the
    quantity of <value expr="application.current_item +
        (num_of_items != 1 ? 's' : '')"/>
    to add, say no or press the zero key on your phone.

    <assign name="global_help" expr="false" />
    <throw event="help" />
</catch>
</field>
<filled namelist="num_of_items confirm">
    <!-- Only whole positive quantities will be accepted -->
    <!-- Make sure the user entered an integer -->
    <var name="is_fraction"
        expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
    <var name="is_negative" expr="num_of_items < 0" />

    <if cond="confirm == false">
        <clear namelist="num_of_items confirm" />
    <elseif cond="num_of_items == 0"/>
        <prompt>
            You have chosen not to add any
            <value expr="application.current_item + 's'" />
            to your shopping cart.
        </prompt>
        <goto next="WelcomeMenuProp.vxml#what_next" />
    <elseif cond="is_fraction || is_negative" />
        <throw event="error.invalid_quantity" />
    <else/>
        <if cond="application.current_item=='book'" >
            <assign name="application.total_books"
                expr="application.total_books
                    + Number(num_of_items)" />
        <else/>
            <assign name="application.total_shirts"
                expr="application.total_shirts
                    + Number(num_of_items)" />
        </if>
        <submit next=".//servlets/CalculateCostAdd"
            namelist="num_of_items" fetchtimeout="3s" />
    </if>
</filled>
</form>
</vxml>

```

在上面的PurchaseStuff程序举例中，你将变量 num_of_items 赋予应用程序变量 num_current_order，并且在CalculateCost文档中引用了 num_current_order。在这个程序例子中，你通过 submit 元素中的 namelist 属性，将 num_of_item 直接传递给 java 小服务应用程序

CalculateCostAdd。Methed属性使用其默认值 get，表示包含 num_of_item 的查询字符串将是 URL 的一部分。默认值application/x-www-form-urlencoded 用于属性enctype。我们将在第12章中介绍如何引用下一个VoiceXML页的变量。

当需要将信息传递给服务器端的脚本文件时，或者像本程序例子中的情况那样，当下面的VoiceXML文档是需要引用被传递的变量的惟一文档时，submit 元素可以用来取代goto 元素。传递num_of_item后，就可以从根文档中删除current_num_order，并且在PurchaseStuff中删除将num_of_items 赋予current_num_order 的assign 元素。

10.7 完成应用程序中VoiceXML部分的开发

在程序清单10-7的示例程序中，实现的功能是列出用户购物车中的所有商品和合计费用。新的ListItems.vxml文档是通过 WelcomeMenuProp.vxml中的what_next菜单来调用的。

程序清单10-7 ListItems.vxml 和CalculateCostList.vxml

```

<?xml version="1.0"?>
<!-ListItems.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly" />
    <meta name="copyright"
        content="Sams Publishing 2001" />
    <meta name="description"
        content="List the content of the shopping cart,
        to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
    <meta name="robots" content="index,follow" />
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST" />
    <meta name="maintainer" content="johnoreilly@lucent.com" />
    <meta name="application" content="Book Store" />

    <form id="output_total">
        <property name="fetchtimeout" value="2s" />
        <block>
            <audio src="cartcontains.wav">
                Your shopping cart now contains
            </audio>
            <value expr="String(Math.round(application.total_books))" />

            <prompt cond="application.total_books == 1">
                <audio src="book.wav">book</audio>
            </prompt>
            <prompt cond="application.total_books != 1">
                <audio src="books.wav">books</audio>
            </prompt>

            <audio src="and.wav"> and </audio>
            <value expr="String(Math.round(application.total_shirts))" />
        </block>
    </form>

```

```

        <else/>
            <assign name="application.total_tshirt_cost"
                   expr="Number(discount.price)" />
        </if>
        <!-- Keep track of the cost of the books added -->
        <var name="old_cost"
              expr="application.total_order_cost" />
        <assign name="application.total_order_cost"
              expr="application.total_book_cost
                    + application.total_tshirt_cost" />
        <!-- Cost of the books added = new total - old total. -->
        <assign name="added_cost"
              expr="application.total_order_cost - old_cost" />
        <goto next="#output_total" />
    </filled>
</subdialog>
</form>

<form id="output_total">
    <property name="fetchtimeout" value="2s"/>
    <block>
        <audio src="youhave.wav"> You have </audio>
        <audio src="added.wav"> added </audio>
        <value expr="String(Math.round(added_items))"/>

        <!-- output book, books, tshirts, or tshirts -->
        <prompt cond="added_items == 1 &&
                      application.current_item == 'book'">
            <audio src="book.wav">book</audio>
        </prompt>
        <prompt cond="added_items != 1 &&
                      application.current_item == 'book'">
            <audio src="books.wav">books</audio>
        </prompt>
        <prompt cond="added_items == 1 &&
                      application.current_item == 'shirt'">
            <audio src="tshirt.wav">t-shirt</audio>
        </prompt>
        <prompt cond="added_items != 1 &&
                      application.current_item == 'shirt'">
            <audio src="tshirts.wav">t-shirts</audio>
        </prompt>

        <audio src="to.wav">to </audio>
        <audio src="yourcart.wav">your shopping cart. </audio>
        <audio src="thecost.wav">The cost of the </audio>
        <value expr="String(Math.round(added_items))"/>
        <audio src="added.wav">added</audio>
        <!-- output book, books, tshirts, or tshirts -->
        <prompt cond="added_items == 1 &&
                      application.current_item == 'book'">
            <audio src="book.wav">book</audio>

```

```

</prompt>
<prompt cond="added_items != 1 && application.current_item == 'book'>
    <audio src="books.wav">books</audio>
</prompt>
<prompt cond="added_items == 1 && application.current_item == 'shirt'>
    <audio src="tshirt.wav">t-shirt</audio>
</prompt>
<prompt cond="added_items != 1 && application.current_item == 'shirt'>
    <audio src="tshirts.wav">t-shirts</audio>
</prompt>

<audio src="is.wav">is </audio>
<value expr="added_cost" class="currency" />
<goto next="ListItems.vxml" fetchtimeout="4s" />
</block>
</form>
</vxml>

```

若要实现 ListItems.vxml，你只需要将用于列出购物车中的商品和购物车中商品的合计费用的函数从CalculateCost文档中的 output_total窗体移动到 ListItems.vxml 文档中即可。现在，当 CalculateCostList.vxml 程序需要一个函数时，它只需转到 ListItems.vxml 文档即可。

在程序清单10-8中，添加了该函数，从而使用户可以从他的购物车中删除有关的商品。该函数可以通过WelcomeMenuProp.vxml程序中的 what_next 菜单来访问。

程序清单10-8 RemoveItem.vxml, RemoveStuff.vxml 和 CalculateCostRemove.vxml

```

<?xml version="1.0"?>
<!-- RemoveItem.vxml -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
        content="Sams Publishing 2001"/>
    <meta name="description"
        content="Select Item to remove, to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <meta name="maintainer" content="johnoreilly@lucent.com"/>
    <meta name="application" content="Book Store"/>

    <form id="pick_item">
        <field name="item">
            <prompt>
                <enumerate>
                    To remove <value expr="_prompt"/>
                    from your shopping cart, say <value expr="_prompt"/>

```

```

        or press <value expr="_dtmf"/>.
    </enumerate>
</prompt>
<option dtmf="1" value="book"> books </option>
<option dtmf="2" value="shirt"> tee shirts </option>
<filled>
    <assign name="application.current_item" expr="item" />
    <goto next="RemoveStuff.vxml"/>
</filled>
<help cond="global_help">
    You are currently selecting an item to remove from your
    shopping cart. Say the name of, or press the phone key
    of the item you would like to remove.

    <assign name="global_help" expr="false" />
    <throw event="help" />
</help>
</field>
</form>
</vxml>

<?xml version="1.0"?>
<!--RemoveStuff.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Remove j items to a shopping cart,
        to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <form id="collect_items">
        <property name="bargein" value="true"/>
        <property name="timeout" value="4s"/>
        <catch event="error.invalid_quantity">
            I'm sorry, <value expr="num_of_items" /> is not a valid entry.
            Order quantities can not be fractions or negative.
            Let's try again.
            <clear namelist="num_of_items confirm" />
            <goto nextitem="num_of_items" />
        </catch>

        <catch event="error.too_many">
            I'm sorry, the <value expr="num_of_items" />
            <value expr="application.current_item +
                (num_of_items != 1 ? 's' : '')"/>
            you have asked to remove from your shopping cart is more than the
            <value expr="(application.current_item=='book' ?
                application.total_books :
                application.total_shirts)" />
            <value expr="application.current_item +

```

```

        (num_of_items != 1 ? 's' : '')"/>
already in your shopping cart.
Let's try again.
<clear namelist="num_of_items confirm" />
<goto nextitem="num_of_items" />
</catch>

<field name="num_of_items" type="number">
  <prompt>
    How many <value expr="application.current_item + 's'" />
    would you like to remove from your shopping cart?
  </prompt>
  <catch event="help" cond="global_help">
    Say or enter using the number keys on your phone the quantity
    of <value expr="application.current_item + 's'" /> you wish to
    remove from your shopping cart.

    <assign name="global_help" expr="false" />
    <throw event="help" />
  </catch>
</field>

<field name="confirm" type="boolean?y=1;n=0">
  <prompt>
    You have asked to remove
    <value expr="num_of_items" /> <!-- number of items -->
    <!-- singular or plural of the item being added -->
    <value expr="application.current_item +
      (num_of_items != 1 ? 's' : '')"/>
    from your shopping cart. Is this correct?
  </prompt>
  <catch event="help" cond="global_help">
    If the quantity you entered is correct, say yes or press the
    one key on your phone.
    If you would like to reenter the quantity of
    <value expr="application.current_item +
      (num_of_items != 1 ? 's' : '')"/> to remove,
    say no or press the zero key on your phone.
    <assign name="global_help" expr="false" />
    <throw event="help" />
  </catch>
</field>
<filled namelist="num_of_items confirm">
  <!-- Only whole positive quantities will be accepted -->
  <!-- Make sure the user entered an integer -->
  <var name="is_fraction"
    expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
  <var name="is_negative" expr="num_of_items < 0" />

  <if cond="confirm == false">

```

```

<clear namelist="num_of_items confirm" />
<elseif cond="num_of_items == 0"/>
<prompt>
    You have chosen not to remove any
    <value expr="application.current_item + 's'" />
    from your shopping cart.
</prompt>
<goto next="WelcomeMenuProp.vxml#what_next" />
<!-- Make sure the user entered an integer -->
<elseif cond="is_fraction || is_negative" />
    <throw event="error.invalid_quantity" />
<!-- Make sure the user does not try to remove more
    than he or she has -->
<elseif cond="num_of_items &gt;
    (application.current_item=='book' ?
        application.total_books :
        application.total_shirts)" />
    <throw event="error.too_many" />
<else/>
    <if cond="application.current_item=='book'" >
        <assign name="application.total_books"
            expr="application.total_books -
                Number(num_of_items)" />
    <else/>
        <assign name="application.total_shirts"
            expr="application.total_shirts
                - Number(num_of_items)" />
    </if>
    <submit next=".//servlets/CalculateCostRemove"
        namelist="num_of_items" fetchtimeout="3s" />
    </if>
</filled>
</form>
</vxml>

<?xml version="1.0"?>
<!--.CalculateCostRemove.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly" />
    <meta name="copyright"
        content="Sams Publishing 2001" />
    <meta name="description"
        content="Calculate the cost of an order, to access
        =>call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
    <meta name="robots" content="index,follow" />
    <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST" />
    <meta name="maintainer" content="johnoreilly@lucent.com" />
    <meta name="application" content="Book Store" />
    <var name="removed_items" expr="application.num_current_order" />

```

```

<form id="calculate_price">
    <subdialog name="discount"
        src=".//CalculateDiscount.vxml#calculate_discount">
        <param name="quantity"
            expr="application.current_item=='book' ?
                application.total_books :
                application.total_shirts" />
        <param name="unit_price"
            expr="application.current_item=='book' ?
                application.unit_book_price :
                application.unit_tshirt_price" />
        <param name="isCoupon"
            expr="application.current_item=='book' ?
                application.has_coupon : false" />
        <filled>
            <if cond="application.current_item == 'book'">
                <assign name="application.total_book_cost"
                    expr="Number(discount.price)" />
            <else/>
                <assign name="application.total_tshirt_cost"
                    expr="Number(discount.price)" />
            </if>
            <assign name="application.total_order_cost"
                expr="application.total_book_cost
                    + application.total_tshirt_cost" />
            <goto next="#output_total" />
        </filled>
    </subdialog>
</form>

<form id="output_total">
    <property name="fetchtimeout" value="2s" />
    <block>
        <audio src="youhave.wav"> You have </audio>
        <audio src="removed.wav"> removed </audio>
        <value expr="String(Math.round(removed_items))" />

        <!-- output book, books, tshirts, or tshirts -->
        <prompt cond="removed_items == 1 &&
            application.current_item == 'book'">
            <audio src="book.wav">book</audio>
        </prompt>
        <prompt cond="removed_items != 1 &&
            application.current_item == 'book'">
            <audio src="books.wav">books</audio>
        </prompt>
        <prompt cond="removed_items == 1 &&
            application.current_item == 'shirt'">
            <audio src="tshirt.wav">t-shirt</audio>
        </prompt>
        <prompt cond="removed_items != 1 &&
            application.current_item == 'shirt'">

```

```

        <audio src="tshirts.wav">t-shirts</audio>
    </prompt>
    <audio src="from.wav">from </audio>
    <audio src="yourcart.wav">your shopping cart. </audio>
    <goto next="ListItems.vxml" fetchtimeout="4s" />
</block>
</form>
</vxml>

```

RemoveItem_vxml程序对AddItemEvent.vxml进行了修改，以提示用户输入他想从购物车中删除的商品，而不是输入他要添加给购物车的商品。这个修改很小，只不过是用词组”remove form”取代”add to”，并且更新下一页，使之指向RemoveStuff.vxml。

RemoveStuff.vxml对PurchaseStuff.vxml进行了修改，以提示用户输入他想从购物车中删除的商品的数量，而不是输入用户想添加的商品的数量。这些修改包括用”remove form”取代”add to”，更新下一页，使之指向CalculateCostRemove小服务程序，并且添加了一个事件处理程序，以便处理用户试图删除的图书或T恤衫的数量大于购物车中的数量这样的问题。

CalculateCostRemove.vxml对CalculateCostProp.vxml进行了修改。CalculateCostRemove.vxml程序调用CalculateDiscount子对话框，以便计算新的订购商品的费用。接着，该应用程序通知用户，他要求删除的这个数量的图书或T恤衫已被删除。最后，CalculateCostRemove.vxml转至ListItems.vxml。

10.8 小结

本章介绍了剩余的几个VoiceXML元素，即record、tranfer、script、property、object和subject。在本章中我们完成了在线书店应用程序的VoiceXML部分的开发。在第12章中，在线书店应用程序将添加相应的函数，以便上载本章的“record元素”这一节中录制的说明文件。另外，我们还将使用服务器端运行的编程语言来实现“submit元素”这一节介绍的CalculateCostAdd.java程序。

第11章 VoiceXML中的ECMAScript

ECMAScript是VoiceXML语言支持的客户机端脚本编程语言。ECMAScript 是基于 JavaScript1.1的脚本语言。在在线书店应用程序中，通过介绍变量的声明、expr 属性和script 元素，我们已经讲过ECMAScript代码的一些例子。

VoiceXML中的变量是根据ECMAScript脚本语言的技术规范来分类的。当用在expr属性中时，ECMAScript提供了一种机制，以便对不同类型的变量的运行进行评估。当用在script元素中时，ECMAScript为程序员提供了传统的编程功能，比如在VoiceXML语言中找不到的loops、case等语句。ECMAScript还提供了许多用户定义的对象、函数、类型转换、数组、条件逻辑等非常有用的特性，还提供了一个函数库，可用于String、Math、Date、Number、Boolean、RegExp和Array等对象。

本章主要有两个目的。首先，为你提供关于ECMAScript的足够信息，使你在编写VoiceXML应用程序时能够充分利用它的各项功能。第二，介绍ECMAScript在 VoiceXML应用程序开发中的作用。为了达到二个目的，我们提供了许多程序举例来说明我们的在线书店应用程序是如何使用ECMAScript脚本语言的。

11.1 ECMAScript注释

ECMAScript 注释遵循标准的Java 约定，通常用两种方法来表示。第一种方法是注释文本的开头使用两个斜杠(//)，结尾是换行符：

```
//comment text
```

第二种方法是注释文本的开头使用一个斜杠和一个星号(*),结尾的字符与开头相反(*):

```
/* comment text */
```

11.2 变量的类型

在第8章的“变量”一节中我们讲过，ECMAScript支持6个标准的变量类型。我们已经在应用程序中使用过每个类型的变量。当你在 ECMAScript 中定义一个变量时，并不需要设定它的变量类型。变量的类型是由赋予它的值或者在undefined的情况下由没有赋予它的值来确定的。下面列出了变量的类型：

- 未定义的变量 (Undefined) ——被赋予一个值之前变量的默认值。例如：

```
<var name="added_cost" />  
<field name="item">
```

- 空值变量(Null)——只能拥有值 null 的变量。

```
<assign name="state" expr="null"/>
```

- 布尔值变量(Boolean)——只能拥有值true或false。例如:

```
<var name="has_coupon" expr="false" />
```

- 字符串变量(String)——封装在单引号中的字符串。例如:

```
<var name="current_item" expr="'book'" />
```

- 数字变量(Number)——任何正或负的整数或小数。例如:

```
<var name="total_order_cost" expr="0" />
```

```
<var name="unit_book_price" expr="24.95" />
```

- 对象变量(Object)——一组无序的属性。每个属性均由名字、值和一组属性组成。ECMAScript既支持用户定义的对象，也支持若干内置的对象。

11.3 用户定义的对象

第10章的“object元素”一节中介绍了一个建立对象并且通过 object 元素引用它的属性的程序举例:

```
<object name="card" classid="java:voicexml.objects.ValidateCreditCard.class">
    <prompt>
        One moment while we validate your credit card.
    </prompt>
    <param name="amount" expr="application.total_order_cost"/>
    <param name="company" expr="credit_card_company"/>
    <param name="card_number" expr="credit_card_number"/>
    <param name="expiration_month" expr="exp_month"/>
    <param name="expiration_year" expr="exp_year"/>
    <filled>
        <prompt>
            Your credit card has been <value expr="card.status"/>
            <!--approved or denied-->
            Please hold while we transfer you to the operator.
        </prompt>
        <goto next="CallOperator.vxml" />
    </filled>
</object>
```

对象也可在script元素中声明和引用。下面这个示例程序建立了一个credit_card_transaction 对象，其属性包括 amount、company、card_number、expiration_month 和 expiration_year，每个属性均被赋予一个值:

```
<?xml version="1.0"?>
<!--CreditCardObject.vxml-->
<!--This example demonstrates the creation of a credit card transaction
     object-->
<vxm version="1.0">
    <form>
        <block>
            <script>
                credit_card_transaction = new Object();
                credit_card_transaction.amount = 24.95;
                credit_card_transaction.company = 'BMEX';
```

```

        credit_card_transaction.card_number = '5555-5555-5555-5555';
        credit_card_transaction.expiration_month = 12;
        credit_card_transaction.expiration_year = 2002;
    </script>
</block>
</form>
</vxml>
```

对象属性可以是方法。下面的示例创建了名为expired的方法并将它指定给对象credit-card-transaction。expired将当前月和年的值作为参数，如果事务中的信用卡到期，该参数返回true，否则返回false：

```

<?xml version="1.0"?>
<!--CreditCardObjMeth.vxml-->
<!--This example demonstrates object methods-->
<form>
    <block>
        <script>
            credit_card_transaction = new Object();
            credit_card_transaction.amount = 24.95;
            credit_card_transaction.company = 'BMEX';
            credit_card_transaction.card_number = '5555-5555-5555-5555';
            credit_card_transaction.expiration_month = 12;
            credit_card_transaction.expiration_year = 2002;

            function expired(current_month, current_year)
            {
                current = current_year * 100 + current_month;
                expiration = this.expiration_year * 100
                    + this.expiration_month;
                if (current > expiration)
                {
                    return true;
                }
                else
                {
                    return false;
                }
            }
            credit_card_transaction.expired = expired;
        </script>
        <if cond="credit_card_transaction.expired(09, 2001)">
            I'm sorry your credit card has expired.
        <else/>
            Your credit card is still valid.
        </if>
    </block>
</form>
</vxml>
```

11.4 内置对象

ECMAScript提供了若干个内置类型的变量，包括 Array、Date、Math、Function、Boolean、

Number、String和RegExp等。下面我们要详细介绍若干个这样的对象。

11.4.1 Array对象

最有用的ECMAScript对象之一是 Array对象。Array（数组）是通过下面两种构造代码来建立的：

```
arrayObject = new Array(item0, item1, ... itemN)
```

或者

```
arrayObject = new Array(length)
```

其中 item0、item1、...itemN 是指数组中的元素，length是数组的初始长度。

若要对数组的元素进行引用和赋值，方法是设定数组的名字，然后设定包含在括号内的被引用的数组元素的指数。ECMAScript中的数组指数从0开始。数组达到长度可以作为对象属性来访问：

```
arrayLength = arrayObject.length;
```

数组也有若干个方法：

- Concat (array2, array3, ...arrayN) ——返回两个或多个数组的组合。
- Join () ——返回一个包含数组的所有元素的字符串。
- Pop () ——删除数组的最后一个元素，然后返回该元素。
- Push (item1, item2, ...itemN) ——将一个或多个元素添加给数组，然后返回添加的最后一个元素。
- Reverse () ——将数组的各个元素的顺序颠倒。
- Shift () ——删除数组的第一个元素，然后返回该元素。
- Slice (begin, ?) ——返回的数组包含了其指数大于或等于 begin参数但小于 end 参数的所有元素。End参数是视情况而定的，如果它没有被设定，那么其指数大于或等于 begin参数的所有元素都将包含在返回的数组中。
- Splice (begin, number, item1, item2, ...itemN) ——删除从指数begin开始的各个number项，再将元素 item1, item2, ...itemN 添加给以指数begin开始的数组。返回包含被删除的数组。
- Sort () ——对数组的各个元素进行排序。
- Unshifted (item1, item2, ...itemN) ——将一个或多个元素添加到数组的前面，然后返回新长度的数组。

11.4.2 Date对象

使用 Date对象，应用程序开发人员就可以在VoiceXML应用程序中进行日期和时间的设置操作。Date对象用于存放月、日、年、时、分和秒。它配有4个构造代码。第一个构造代码不带任何参数，即dateObject=new Date ()。这个构造代码用于建立使用当前月、日、年、时、分和秒的Date对象。该构造代码用在在线书店示例程序的CheckOutScript.vxml文件中，即“date = new Date ()”；。第二个Date构造代码用一个代表日期的String作参数来建立日期：

```
dateObject = new Date("Month day, year hours:minutes:seconds")
```

请注意，在这个字符串参数中，月是个英文单词，而不是个数字。如果没有设定时、分、秒，那么它们将被赋予一个值0。第3个构造代码使用3个整数值作为年、月、日：

```
dateObject = new Date(year, month, day)
```

时、分、秒均设置为0。第4个构造代码使用6个整数值作为年、月、日、分、秒：

```
dateObject = new Date(year, month, day, hour, minute, second)
```

Date对象使用 get和 set方法来读取和设置各个组成部分的值。我们将get方法用于在线书店示例程序的CheckOutScript.vxml文件中的月和日的检索和赋值：

```
month = date.getMonth();
day = date.getDate();
```

表11-1显示了用于Date对象的get和set方法，简单描述了它们设置和返回的属性，以及这些属性的值的有效范围。

表11-1 用于Date对象的get和 set 方法

函数	描述	有效值
GetSeconds, SetSeconds	Date中的秒	整数：0-59
GetMinutes, SetMinutes	Date中的分	整数：0-59
GetHours, SetHours	Date中的时	整数：0-23
GetDay,	星期几	整数：0（星期日）—6 （星期六）
GetDate, SetDate	一个月中的第几日	整数：1-31
GetMonth , SetMonth	Date中的月份	整数：1-11
GetYear, SetYear	Date中的从1900年算起的年份	所有整数
GetTime, SetTime	Date中的从1970年1月1日算起的毫秒数	所有整数

如果输入的整数值不在有效范围之内，设置的Date属性就会绕回，下一个最大的或最小的Date 属性将自动被更新。例如，如果月份是7，而执行的代码是object .getDate (34)，结果month更新为8，而 date 则绕回，被更新为3。

11.4.3 Math对象

使用Math对象，应用程序开发人员能够对数字执行各种数学运算操作。下面的列表显示了Math对象使用的各种方法和对每个函数返回的结果的描述。所有方法都是静态的，也就是说不必创建 Math 对象来调用这些方法：

- Abs (number) ——number的绝对值。
- Asin (radians)、acos (radians)、atan (radians) ——反三角函数。
- Ceil (number) ——大于或等于number的最小整数。
- Exp (base_e_number)、log(base_e_number)——对数函数。
- Floor(number)——小于或等于number的最大整数。
- Max (number1, number2) ——取两个数中较大的数。
- Min (number1, number2) ——取两个数中较小的数。

- Pow (base, exponent) ——base的 exponent次方。
- Random () ——0与1之间的伪随机数。
- Round (number) ——规整为最近整数。
- Sin (radians)、cos (radians)、tan (radians) ——三角函数。
- Sqrt (number) ——number的平方根。

我们在整个在线书店应用程序中都使用了 Math对象和它的方法。例如，在 PurchaseStuffSubmit.vxml 程序中，我们使用ceil和Floor 函数来确保变量是个整数：

```
<var name="is_fraction"
      expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
```

另外，我们还在CalculateCostProp.vxml程序中使用 round 函数，使数字值圆整为一个整数：

```
<value expr="String(Math.round(added_items))"/>
```

11.4.4 String对象

使用String对象，应用程序开发人员就能够使用字符串并且对字符串进行相应的操作。String的属性 length是指字符串的长度。String 对象也拥有若干种方法。下面列出了String的方法和每个函数返回的结果：

- CharAt (index) ——字符串中index 位置上的字符。
- CharCodeAt (index) ——index位置上的字符代码。
- IndexOf (string, [start]) ——从索引start开始的 string第一次出现的位置，如果 start 没有设定，搜索便从0开始。如果没有找到string，便返回-1。
- LastIndexOf (string, [start]) ——从索引start开始的 string最后一次出现的位置，如果 start 没有设定，搜索便从0开始。如果没有找到string，便返回-1。
- Replace (/old/, new) ——包含old的子字符串的所有字符串都被字符串 new 所取代。
- Split (separator) ——使用separator将字符串分割成许多子字符串而获得的字符串数组。
- Substring (begin, length) ——从位置begin开始到位置 begin+length结束的子字符串。
- Substring (begin, end) ——从位置begin 开始到位置end 结束的子字符串。
- ToLowerCase () ——全部变为小写的字符串。
- ToUpperCase () ——全部变为大写的字符串。

11.4.5 函数与 Function对象

ECMAScript的函数可以用两种方法来声明。一种方法是通过建立 Function 对象来声明函数，另一种方法也是首选方法，它使用关键字Function来声明函数。用于Function对象的构造代码是：

```
functionObject = new Function([arg1, arg2, ... argN], function_body)
```

其中arg1、arg2、...argN 都是函数的参数，它们的使用是根据情况而定的。function_body 是一个string，用于设定调用函数时要执行的代码。下面的代码段用于建立一个简单的函数对象，以便将货运费用添加给购物订单。然后该函数被调用时，其结果被赋予给一个变量：

```
total = new Function("order_cost",
```

```

        "shipping",
        "return (order_cost + shipping)");
total_cost = total(7, 1);

```

函数还可以用function关键字来声明：

```

function functionObject([arg1, arg2, ... argN]) {function_body}
total函数可以用下面这个代码格式重新编写：

```

```

function total(order_cost, shipping)
{
    return (order_cost + shipping);
}
total_cost = total(7, 1);

```

函数也可以用来建立对象的构造代码。用于建立本章前面部分中介绍的对象credit_card_transaction的语法如下：

```

function credit_card_transaction(amount,
                                  company,
                                  card_number,
                                  expiration_month,
                                  expiration_year)
{
    this.amount=amount;
    this.company=company;
    this.card_number=card_number;
    this.expiration_month=expiration_month;
    this.expiration_year=expiration_year;
}

```

若要建立一个对象实例，只需使用必要的参数来调用该构造代码即可：

```

transaction_instance = new credit_card_transaction(24.95,
                                                    'BMEX',
                                                    '5555-5555-5555-5555',
                                                    12,
                                                    2002);

```

11.5 运算符

与大多数语言一样，ECMAScript也支持数学运算符，条件运算符，赋值运算符，布尔运算符和其他运算符。ECMAScript不是一种分类十分严格的语言，也就是说任何类型的变量都可以被赋予任何类型的其他变量。表11-2至11-7列出了ECMAScript支持的各种类型的变量。

表11-2 二进制数学运算符

运算符	说 明
$x + y$	加
$x - y$	减
$x * y$	乘
x / y	除
$x \% y$	求模（除法运算的余数）

表11-3 一元数字运算符

运算符	说 明
$x++$ 或 $++x$	将 x 递增1
$x--$ 或 $--x$	将 x 递减1
$-x$	x 求反

表11-4 条件运算符

运算符	说 明
$x == y$	如果 x 等于 y , 返回true, 否则返回false
$x != y$	如果 x 不等于 y , 返回true, 否则返回false
$x === y$	如果 x 等于 y , 并且 x 和 y 属于同一类型, 返回true, 否则返回false
$x !== y$	如果 x 不等于 y , 并且 x 和 y 不属于同一类型, 返回true, 否则返回false
$x > y$	如果 x 大于 y , 返回true, 否则返回false
$x >= y$	如果 x 大于或等于 y , 返回true, 否则返回false
$x < y$	如果 x 小于 y , 返回true, 否则返回false
$x <= y$	如果 x 小于或等于 y , 返回true, 否则返回false

表11-5 赋值运算符

运算符	说 明
$x += y$	等价于 $x = x + y$
$x -= y$	等价于 $x = x - y$
$x *= y$	等价于 $x = x * y$
$x /= y$	等价于 $x = x / y$
$x \% y$	等价于 $x = x \% y$

表11-6 布尔运算符

运算符	说 明
$x \&& y$	如果 x 和 y 是true, 该表达式的计算结果为true, 否则计算结果为false
$x y$	如果 x 或 y 是true, 该表达式的计算结果为true, 否则计算结果为false
$! x$	如果 x 是false, 该表达式的计算结果为true, 否则计算结果为false

表11-7 其他运算符

运算符	说 明
$Condition ? if_true_code : if_false_code$	等价于 <code>if (condition)</code>

(续)

运算符	说 明
Instanceof	{ if_true_code } else { if_false_code }
Typeof	如果某个对象属于既定的对象类型,返回true,否则返回false 返回变量的类型

11.6 类型转换

使用ECMAScript,可以进行对象类型的转换,或者将一种对象的类型转换成另一个类型。对象类型转换的约定方式是NewObjectType (Object_instance)。在线书店应用程序中我们已经若干次使用了类型转换方法。例如,由于将类型为 Number 的变量与类型为String 的变量相加的结果将产生一个类型为String的变量,因此变量 num_of_items必须从 string的变量转换成 Number ,以确保能够为PurchaseStuffSubmit.vxml 中的total_books 获得正确的结果:

```
<assign name="application.total_books"
       expr="application.total_books + Number(num_of_items)" />
```

在CalculateCostProp.vxml程序的下行代码中:

```
<value expr="String(Math.round(added_items))"/>
```

使用String转换方法,用round () 确保返回整数作为输出。

11.7 控制程序的执行

ECMAScript提供了许多非常有用的方法。以便对条件逻辑和循环进行编码。条件逻辑可以使用if/else或case语句来表示。ECMAScript允许建立若干类型的循环,包括for循环、do...while循环、while循环、以及 for...in循环。最后,with命令可以方便地进行与单个对象的语句进行操作。

11.7.1 if/else语句

与VoiceXML一样, ECMAScript将if/else if/else 命令用于条件逻辑。ECMAScript 中的if/else if/else语句的语法如下:

```
if (condition)
{
    if code
}
else if (condition)
{
```

```

        else if code
    }
else
{
    else code
}

```

11.7.2 case语句

当一个表达式的值被多次判别时，case语句提供一种更加紧凑的编写条件逻辑的方法。case语句的语法如下：

```

switch (expression)
{
    case value1:
        code;
        break;
    case value2:
        code;
        break;
    default:
        code;
}

```

ECMAScript的case语句出现在 CheckOutScript.vxml程序中。该程序文档使用 case语句，根据变量month的值来设置要输出的月份：

```

switch (month)
{
    case 0:
        spoken_month = "January";
        break;
    case 1:
        spoken_month = "February";
        break;
    case 2:
        spoken_month = "March";
        break;
    case 3:
        spoken_month = "April";
        break;
    case 4:
        spoken_month = "May";
        break;
    case 5:
        spoken_month = "June";
        break;
    case 6:
        spoken_month = "July";
        break;
    case 7:
        spoken_month = "August";
}

```

```

        break;
    case 8:
        spoken_month = "September";
        break;
    case 9:
        spoken_month = "October";
        break;
    case 10:
        spoken_month = "November";
        break;
    case 11:
        spoken_month = "December";
        break;
}

```

11.7.3 循环

让我们回顾一下第8章中的“循环”一节的内容，在该节中，我们在VoicXML中建立了一个简单的循环：

Pseudo code:

```

i=0
stop = 5
counter = 1

while i < stop do
    counter = counter * 2
    i++
end while

```

VXML code:

```

<?xml version="1.0"?>
<!--Loops.vxml
<!--Demonstrates a while loop -->
<vxml version="1.0" >
<form>
    <var name="stop" expr="5" />
    <var name="i" expr="0" />
    <var name="counter" expr="1" />
    <block name="loop">
        <if cond="i < stop">
            <assign name="i" expr="i+1" />
            <assign name="counter" expr="counter * 2" />
            <clear namelist="loop"/>
        </if>
    </block>
    <!-- The loop is executed 5 times so counter is 32 -->
</form>
</vxml>

```

你将会发现，VoicXML中的这个循环代码远远不如伪代码版本那样自然。如果能够访问

VoiceXML程序中的 ECMAScript，那么应用程序开发人员就能够访问预定义的循环结构，比如 for循环、do...while 循环、while循环和for...in循环。如果必须有一个循环，而该循环的主体并不需要执行任何VoiceXML元素，那么就应该使用封装在VoiceXML 中的ECMAScript循环，而不使用只有 VoiceXML元素的循环。

下面的示例代码说明了如何使用预定义的 ECMAScript循环 for、do...while和 while 来建立以前的循环：

```
//for loop
for (i = 0; i < stop; i++)
{
    counter=counter * 2;
}

//do..while loop
do
{
    counter = counter * 2;
    i++;
} while (i < stop)

//while loop
while (i < stop)
{
    counter=counter * 2;
    i++;
}
```

for...in循环用于处理某个对象的所有属性和某个数组的所有元素。下面的示例代码建立了一个credit_card_transaction对象，使用for...in 循环建立一个string，它配有该对象的所有属性和值，并且输出该string：

```
<?xml version="1.0"?>
<!--CreditCardForIn.vxml-->
<!--This example demonstrates the for in loop-->
<vxml version="1.0">
    <form>
        <block>
            <script>
                credit_card_transaction = new Object();
                credit_card_transaction.amount = 24.95;
                credit_card_transaction.company = 'BMEX';
                credit_card_transaction.card_number = '5555-5555-5555-5555';
                credit_card_transaction.expiration_month = 12;
                credit_card_transaction.expiration_year = 2002;
                output = '';
                for (value in credit_card_transaction)
                {
                    output += value + ':' + credit_card_transaction[value]
                        + ' ';
                }
            </script>
        </block>
    </form>
</vxml>
```

```

    </script>
    <value expr="output" />
  </block>
</form>
</vxml>

```

当执行CreditCardForIn.vxml程序时，便向用户输出下面的 String

```

"amount: 24.95
company: BMEX
card_number: 5555-5555-5555-5555
expiration_month: 12
expiration_year: 2002"

```

有时break和continue命令也可用于循环。Break命令用于结束循环的执行，并且转入代码的下一行。Continue命令用于结束循环的当前迭代，然后转入下一个迭代运行。

11.7.4 with语句

使用 with语句，你可以执行以特定对象为默认对象的一系列语句。例如，下面的代码使用 with语句和credit_card_transaction 对象，该对象带有定义好的属性amount、company、card_number、expiration_month 和expiration_year：

```

with (credit_card_transaction)
{
  amount = 24.95;
  company = 'BMEX';
  card_number = '5555-5555-5555-5555';
  expiration_month = 12;
  expiration_year = 2002;
}

```

11.8 小结

本章讲述了ECMAScript的基本概念和语法，包括注释，变量类型，内置对象及其方法，运算符，类型转换和函数，还介绍了如何通过 if/else语句、case 语句和循环来控制程序的执行。

第12章 高级主题和服务器端的应用程序开发

服务器端的应用程序开发是使你的VoiceXML应用程序成为动态应用程序的另一种方法，它不同于第11章介绍的用ECMAScript来开发VoiceXML应用程序的方法。顾名思义，服务器端的应用程序开发将开发过程转到了Web服务器上，而不是客户机上。

在Web服务器上进行服务器端的应用程序开发是通过开发能够输出 VoiceXML 代码的代码来完成的，然后，输出的VoiceXML代码将传递给 VoiceXML 网关，该网关是你的“客户机”的组成部分。本章中我们介绍的服务器端应用程序开发方法是你可能已经熟悉的开发图形Web 站点的这些方法。

你可以使用各种各样的服务器端应用程序开发语言。常用的每种语言都支持服务器端的程序设计，并且有些语言是专门为这种应用程序开发而创建的。本章我们将介绍两种流行的服务器端应用程序开发使用的语言，并且讲述几个例子，使你能够着手开发服务器端的VoiceXML 应用程序。

本章并不全面介绍某种语言的使用情况，我们的目的是讲述与VoiceXML 相关的一些特定问题，使你在了解这些语言时有一个良好的开端，并且提供一些例子，帮助VoiceXML开发人员解决他们遇到的典型问题。

12.1 语言

下面是对两种流行的服务器端编程语言 Java和 PHP 的描述和使用方法的说明，并且提供了一些使用举例。

我们将重点介绍VoiceXML应用程序开发和将内容传递给客户机的设置要求等方面的具体问题，以及与呼叫方电话一道运行的 VoiceXML网关的情况。有些非常出色的著作都对这些语言和工具进行了详细的介绍。如果你是刚刚开始学习服务器端的应用程序开发，那么你可以从这些著作中挑选一本，作为学习的起点，我们介绍的应用程序开发举例将有助于你进行深入一步的学习。

除非你能够访问你可以进行配置和控制的 Web 服务器，否则你很难通过电话来访问你建立的动态VoiceXML应用程序。虽然通过第3章介绍的语音服务提供商（BeVocal、HeyAnita、Tellme、VoiceGenie 和Voxeo），你可以连接到你在 Web 服务器上建立的VoiceXML内容，比如向ISP申请一个账户，但是很少有 ISP 能够提供我们将要介绍的服务器端的应用程序开发工具。PHP可能是个例外，因为它在语音服务提供商中很流行。

另一种方法是使用IBM公司（网址是`alphaworks.ibm.com/tech/voiceserversdk`）和VoiceGenie公司（网址是`http://developer.voicegenie.com/genietracer.Php`）提供的独立模拟器来

进行开发。虽然你无法使用你的电话来测试动态内容，但是你可以使用这些工具通过 PC 扬声器和话筒来模仿你的应用程序。同样，这种方法的缺点是无法得到你与应用程序交互操作的真实感觉。

12.2 创建VoiceXML应用程序

本节将要介绍用 VoiceXML 进行服务器端应用程序开发的基础知识。我们首先使用 Java 小服务程序和 PHP 来展示服务器端的“Hello World”应用程序，然后我们继续使用在线书店应用程序这个例子来说实际运行的应用程序。

前面讲过，进行服务器端的编程时，你的应用程序是动态的，因为在代码中你可以决定应用程序的输出是什么和 VoiceXML 中的输出采用何种格式，这样，网关就可以将这个输出传递给你的用户。下面两小节将要介绍如何用 Java 和 PHP 来进行这种开发。

12.2.1 Java小服务程序

使用 Java 小服务程序进行服务器端的应用程序开发，必须建立支持这种开发环境的 Web 服务器软件。这里我们使用 Allaire JRun 作为我们的小服务程序引擎。在实际的服务器端应用程序开发环境中，你可能选择运行 Apache 之类的 Web 服务器软件，以便为静态内容提供服务，并且使用 JRun（或类似的小服务程序引擎，如 Tomcat），以便为动态内容提供服务。

1. 建立小服务程序的运行环境

下面的软件是必不可少的：

- Allaire JRun（从网址 www.allaire.com/products/jrun/index.cfm 进行下载）。
- Java 开发工具包（JDK）（从网址 java.sun.com/j2se/ 进行下载）。

你必须使用 JDK 来编译你的小服务程序（如果你没有安装好现成的 Java 运行环境，那么你可以通过小服务程序引擎来运行你的小服务程序）。对于本章中的示例程序，我们将使用太阳公司的 JDK。当你通过安装程序和选定安装目录来运行小服务程序环境后，你必须修改你的运行环境，以便更新你的 PATH，并且设置 CLASSPATH 和 JAVA_HOME。下面是个示例批处理文件，它可以通过命令行来运行，以便执行上述设置：

```
env.bat
REM jdkinstalldir is where you installed the jdk, i.e., c:\jdk1.3.1
set JAVA_HOME=<jdkinstalldir>
set PATH=%JAVA_HOME%\bin;%PATH%
set CLASSPATH=.
```

在设置你的 CLASSPATH 时，你必须记住它不应该指向以前安装的 JDK 的 classes.zip 文件。这些设置可以人工进行，方法是打开一个显示 DOS 提示符的命令控制台，然后分别输入每一个命令。这些设置也可以适用于你打开的每个 DOS 窗口，方法是永久地修改你的运行环境。为此，请依次选择 Start, Settings, Control Panel，再选定 System。选定 Environment 选项卡（Windows NT 上），或者选定 Advanced 选项卡（Windows 2000 上），然后修改环境设置值，或者将环境设置值输入 env.bat 文件。请注意，当你添加更多的组件时，你必须修改 env.bat 文件，以便反映出它们的路径。

你可以从Allaire公司那里免费得到 JRun 的开发人员版本。若要安装JRun , 请下载并执行 jrun_31_win_us_exe。这样你就可以运行一个标准的 Windows安装过程, 在这个安装过程中, 你可以修改默认的安装目录和程序组。你还必须输入你安装 Java的位置和服务器端口号。当安装完成时, 请访问 <http://localhost:8100> (假定你在安装过程中接受默认端口号的话)。这时, 如果你看到 “Index of /” 和两根横线, 就说明安装取得了成功。

前面讲过, 在实际运行环境中, JRun 和类似的小服务程序引擎可以被添加给Web 服务器软件, 不过这里我们并没有这样做, 目的是为了简明起见。

2. 简单的示例程序

现在你有了一个服务器, 它已经配置成能够支持Java 小服务程序, 因此我们将要用一个简单的示例程序来测试你的配置。首先我们要通过Web浏览器来观察 “Hello World” 应用程序, 然后我们要为语音Web文件做一些小的修改。

首先, 用下面的代码建立一个文件, 名叫 HelloWorld.java:

```
HelloWorld.java

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        // Output HTML
        out.println("<html><head><title>Hello world!</title>");
        out.println("</head><body>");
        out.println("<h1>Hello world!</h1>");
        out.println("<p>This is written from a servlet.</p>");
        out.println("</body></html>");
    }
}
```

现在将该文件编译成一个类文件, 方法是在命令行提示符的后面输入下面的命令:

```
javac HelloWorld.java
```

将产生的类文件放入一个JRun能够为之提供服务的目录, 即 <JRuninstalldir>\servers\default\default-app\WEB-INF\classes。现在你应该能够使用 URL [现在我们对该示例程序文件稍作修改, 使之输出 VoiceXML 而不是 HTML文件:](http://localhost:8100/servlet>Hello World , 通过你的 Web浏览器来访问该文件页。这里我们同样假设你接受默认的服务器端口。</p>
</div>
<div data-bbox=)

```
HelloVoiceWorld.java

import java.io.*;
```

```

import javax.servlet.*;
import javax.servlet.http.*;

public class HelloVoiceWorld extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException
    {
        res.setContentType("text/xml");
        PrintWriter out = res.getWriter();
        out.println("<?xml version=\"1.0\"?> ");
        out.println("<vxm version=\"1.0\">");

        out.println("    <form>");           '
        out.println("        <block>");      '
        out.println("            Hello voice world!");'
        out.println("            Spoken from a servlet.");'
        out.println("        </block>");'
        out.println("    </form>");'
        out.println("    </vxm>");'
    }
}

```

上面建立的两个小服务程序的功能非常有限，不过代码中有两个东西值得注意。首先，对`HttpServletResponse.setContentType`的调用发生了变化，以反映返回的不同输出类型。对于某些VoiceXML网关来说，它必须是“text.vxml”类型。

前面我们讲过，若要运行这个示例程序，你可以有3个选择：

- 使用一个语音服务提供商(VSP)，使你的应用程序能够使用它的在线工具连接到你能够配置和控制的Web服务器。
- 如果你的服务提供商支持小服务程序的开发，那么请以同样的方式使用你的ISP账户。如果没有这样的服务提供商，那么可以了解一些由广告资助的免费Java小服务程序主机，比如www.webappcabaret.com。如果你使用这种方法，那么务必要懂得如何将网幅广告插入你的小服务程序的输出。
- 使用桌面模拟程序，直到你已经准备好运行某个应用程序。

最后，请不同忘记你的Web浏览器。语音应用程序开发的一部分工作就是要了解哪一种应用程序适合语音应用，而调试软件则不属于这些应用程序。在试图使用VSP之前，请按照渐进法，用浏览器逐个测试各个小服务程序。通过访问Web浏览器中你的小服务程序，并且查看其来源，你可以完成大量的VoiceXML应用程序的动态测试。你只需要将Web浏览器指向小服务程序，再提供各种必要的参数，然后查看小服务程序产生的输出（它可以粘贴到一个静态VoiceXML文件中，以便在必要时使用电话对它进行测试）。

3. BookStore示例程序

下面我们继续使用在线书店示例程序，来介绍服务器端的小服务程序开发的某些比较详细的情况。下面这个示例程序展示了如何用小服务程序来生成更加广泛的VoiceXML应用程序。

虽然这个示例程序只使用Java来访问一个输入参数，但是它说明了如何编写冗长的

VoiceXML小服务程序的过程。同时，当我们接下来介绍 PHP 脚本程序开发时，它可以用来进行非常有意思比较。

不管使用何种编程语言，开发带有大量输出的服务器端应用程序使用的最简便的方法是，首先用静态标记语言来编写你所需要的输出，然后使这些输出成为动态输出。在程序清单12-1中，我们就是这样做的，首先编写CalculateCostAdd.vxml程序，然后对这个小服务程序作进一步编码。使用这种渐进法，你就可以在使应用程序变为动态程序之前，首先把静态对话框搞得准确无误。

程序清单12-1 CalculateCostAdd.java

```

import javax.servlet.*;
import javax.servlet.http.*;
import java.text.*;
import java.io.*;
import java.net.*;

public class CalculateCostAdd extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse res )
                      throws ServletException, IOException
    {
        res.setContentType( "text/xml" );
        ServletOutputStream out = res.getOutputStream();

        String num_of_items = req.getParameter("num_of_items");

        out.println("<?xml version=\"1.0\"?> " );
        out.println("<!--CalculateCostAdd.vxml-->");
        out.println("<vxm version=\"1.0\" \"");
        out.println("      application=\"BookStoreRoot.vxml\"");
        out.println("<meta name=\"author\" content=\"John O'Reilly\"");
        out.println("      <meta name=\"copyright\"");
        out.println("      content=\"Sams Publishing 2001\"");
        out.println("      <meta name=\"description\"");
        out.println("      content=\"Calculate the cost of a book order, \"");
        out.println("      to access call (xxx)-xxx-xxxx\"");
        out.println("      <meta name=\"keywords\" content=\"VoiceXML Sams Books\"");
        out.println("      <meta name=\"maintainer\" content=\"johnnyo@sams.net\"");
        out.println("      <meta name=\"robots\" content=\"index,follow\"");
        out.println("      <meta http-equiv=\"Date\"");
        out.println("      content=\"Thu, 7 Aug 2000 20:20:20 EST\"");
        out.println("      <meta name=\"maintainer\" \" ");
        out.println("      content=\"johnoreilly@lucent.com\"");
        out.println("      <meta name=\"application\" content=\"Book Store\"");
        out.println("      <var name=\"added_cost\" />\"");
        out.println("      <var name=\"added_items\" expr=\""+num_of_items+"\" />\"");
        out.println("      <form id=\"calculate_price\">\"");
        out.println("      <subdialog name=\"discount\"\"");
        out.println("      src=\"./CalculateDiscount.vxml#calculate_discount\"");
        out.println("      <param name=\"quantity\"\"");
    }
}

```

```

out.println("           expr=\"application.current_item=='book' ?\"");
out.println("               application.total_books : ");
out.println("               application.total_shirts\" />\"");
<param name=\"unit_price\"\">";
out.println("           expr=\"application.current_item=='book' ?\"");
out.println("               application.unit_book_price : ");
out.println("               application.unit_tshirt_price\" />\"");
<param name=\"isCoupon\"\">";
out.println("           expr=\"application.current_item=='book' ? \"");
out.println("               application.has_coupon : false\" />\"");
<filled>");
out.println("           <if cond=\"application.current_item == 'book'\">>\"");
out.println("               <assign name=\"application.total_book_cost\"\">";
out.println("                   expr=\"Number(discount.price)\"/>\"");
<else/>");
out.println("               <assign name=\"application.total_tshirt_cost\"\">";
out.println("                   expr=\"Number(discount.price)\"/>\"");
</if>");
<!-- Keep track of the cost of the books added -->";
out.println("           <var name=\"old_cost\"\"");
out.println("               expr=\"application.total_order_cost\"/>\"");
<assign name=\"application.total_order_cost\"\">";
out.println("               expr=\"application.total_book_cost\"");
out.println("                   + application.total_tshirt_cost\"/>\"");
<!-- Cost of the books added=new total-old total.-->";
out.println("           <assign name=\"added_cost\"\">";
out.println("               expr=\"application.total_order_cost - \"");
out.println("                   old_cost\"/>\"");
out.println("           <goto next=\"#output_total\" />\"");
out.println("           </filled>\"");
out.println("           </subdialog>\"");
out.println("</form>\"");
out.println("<form id=\"output_total\"\"");
out.println("<property name=\"fetchtimeout\" value=\"2s\"/>\"");
out.println("           <block>\"");
out.println("               <audio src=\"youhave.wav\"> You have </audio>\"");
out.println("               <audio src=\"added.wav\"> added </audio>\"");
out.println("               <value expr=\"String(Math.round(added_items))\"/>\"");
<!-- output book, books, tshirts, or tshirts -->\"");
out.println("               <prompt cond=\"added_items == 1 &&\"");
out.println("                   application.current_item == 'book'\">>\"");
out.println("                   <audio src=\"book.wav\">book</audio>\"");
</prompt>\"");
out.println("               <prompt cond=\"added_items != 1 &&\"");
out.println("                   application.current_item == 'book'\">>\"");
out.println("                   <audio src=\"books.wav\">books</audio>\"");
</prompt>\"");
out.println("               <prompt cond=\"added_items == 1 &&\"");
out.println("                   application.current_item==''shirt'\">>\"");
out.println("                   <audio src=\"tshirt.wav\">t-shirt</audio>\"");
</prompt>\"");
<prompt cond=\"added_items != 1 && \">\"");

```

```

out.println("           application.current_item=='shirt'\">");  

out.println("     <audio src=\"tshirts.wav\">t-shirts</audio>");  

out.println("   </prompt>");  

out.println("   <audio src=\"to.wav\">to </audio>");  

out.println("   <audio src=\"yourcart.wav\">");  

out.println("     your shopping cart. ");  

out.println("   </audio>");  

out.println("   <audio src=\"thecost.wav\">The cost of the </audio>");  

out.println("   <value expr=\"String(Math.round(added_items))\"/>");  

out.println("   <audio src=\"added.wav\">added</audio>");  

out.println("   <!-- output book, books, tshirts, or tshirts -->");  

out.println("   <prompt cond=\"added_items == 1 && \"");  

out.println("     application.current_item == 'book'\">");  

out.println("     <audio src=\"book.wav\">book</audio>");  

out.println("   </prompt>");  

out.println("   <prompt cond=\"added_items != 1 && \"");  

out.println("     application.current_item == 'book'\">");  

out.println("     <audio src=\"books.wav\">books</audio>");  

out.println("   </prompt>");  

out.println("   <prompt cond=\"added_items == 1 && \"");  

out.println("     application.current_item=='shirt'\">");  

out.println("     <audio src=\"tshirt.wav\">t-shirt</audio>");  

out.println("   </prompt>");  

out.println("   <prompt cond=\"added_items != 1 && \"");  

out.println("     application.current_item=='shirt'\">");  

out.println("     <audio src=\"tshirts.wav\">t-shirts</audio>");  

out.println("   </prompt>");  

out.println("   <audio src=\"is.wav\">is </audio>");  

out.println("   <value expr=\"added_cost\" class=\"currency\" />.");  

out.println("   <goto next=\"ListItems.vxml\" fetchtimeout=\"4s\" />");  

out.println(" </block>");  

out.println(" </form>");  

out.println("</vxml>");  

out.close();
}
}

```

请注意这个脚本程序是如何将原始VoiceXML对话框的名字纳入小服务程序源代码中，以便帮助程序调试的。另外，它也保持了小服务程序原来的缩排形式。如果你发现自己正在编写复杂的小服务程序，并且有时使用图形Web浏览器来测试它的输出，那么这是一个非常重要的调试辅助手段，它使你可以避免看到许多没有缩排或者缩排格式不佳的源代码。

12.2.2 PHP脚本程序

正如你在建立Java小服务程序时的情况一样，若要使用PHP来进行服务器端的应用程序开发，你必须建立支持这种开发环境的服务器软件。我们下面介绍的开发环境包括了Apache Web服务器，并且已经安装了PHP处理程序模块。

1. 建立PHP开发环境

下面两个软件是必不可少的：

- Apache Web服务器软件（从网址www.apache.org下载）。
- PHP（从网址www.php.net下载）。

请按上面列出的顺序下载和安装这两个软件。Apache使用一种简单易行的 Windows安装程序。Web服务器的管理是通过一个Windows 程序组来完成的。

若要启动该服务器，请依次选定Start, Programs, Apache httpd Server, Start Apache In Console。若要测试你安装的服务器，请在URL处输入http://localhost，这时你应该看到Apache 的欢迎页面。

当你下载了PHP 二进制文件后，请将该文件解压缩到一个安装目录，然后按照install.txt文件中的说明进行操作：

- 1) 当Apache停止运行时，请编辑httpd.conf文件，添加下面几个代码：

```
LoadModule php4_module <phpinstalldir>/sapi/php4apache.dll
AddType application/x-httdp-php .php4
ScriptAlias /php4/ "<phpinstalldir>"
Action application/x-httdp-php4 "/php4/php.exe"
AddType application/x-httdp-php4 .php
```

- 2) 将下载的文件解压缩后纳入一个安装目录，然后从该目录将文件 php4ts.dll 移动到window/system (32)，覆盖任何老版本的文件。

- 3) 将文件 php.ini-dist 拷贝到你安装Windows的目录下，然后将该文件改名为 php.ini。

- 4) 重新启动apache服务器。

2. 简单的示例程序

现在你已经对web服务器进行了配置，以支持 PHP。下面是一个简单的示例程序，用于测试你所做的配置：

```
HelloWorld.php
<html>
<head><title>Hello World</title></head>
<body><?php echo "Hello World! Written from PHP<p>"; ?></body>
</html>
```

若要给这个示例程序加上VoiceXML运行机制，请看下面的程序：

```
HelloVoiceWorld.php
<?php header("Content-Type: text/xml");?>
<?xml version="1.0"?>
<vxml version="1.0">

<form>
<block>
<?php echo "Hello Voice World! Spoken from PHP";?>
</block>
</form>
</vxml>
```

服务器端的脚本程序源代码包含在<?...?>标记中。我们再一次设置了内容类型，以反映VoiceXML 网关的要求。否则，惟一的变化是它使用了特定的语音标记。

3. BookStore示例程序

请回忆一下程序清单12-1中的Java小服务程序在线商店示例程序，从静态程序页变成动态程

序页的过程，必须修改每个代码行，尽管有些代码行只需稍作修改。这是个非常麻烦而且很容易出错的过程。**PHP**是一种服务器端的嵌入式编程语言。因此它避免了相当多的这种修改过程。

当**Web**服务器用服务器端的脚本程序来处理一个对话框时，你可以将它视为根据需要来调用解释程序。为此，服务器要遍历一个文件，直到它到达一个表示脚本编程语言的标记为止，再通过结束标记来调用代码的解释程序，然后重新恢复标准的标记语言的处理方式，在我们的例子中，这个标记语言就是**VoiceXML**。

观察下面的示例程序，它的明显优点是代码更加容易阅读。它有比较少的代码行涉及两种编程语言，例如：

```
out.println("<vxm version=\"1.0\">");
```

甚至有些代码涉及3种类型的编程语言，例如：

```
out.println("<audio src=\"thecost.wav\">The cost of the </audio>\n        <value expr=\"String(Math.round(added_items))\"/>");
```

虽然它们看起来差别很小，但代码的调试和维护比较困难，因为它必须用若干不同的语言环境来思考问题。

下面的代码段（**CalculateCostAdd.vxml**程序的头几行）说明了**PHP**是如何更快地从静态程序页转换成动态程序页的。由于**PHP**语言具备的嵌入式特性，因此只有动态代码行需要进行脚本编程，其他代码行与静态对话框没有什么变化：

```
CalculateCostAdd.vxml\n<?xml version="1.0"?>\n<vxm version="1.0" application="BookStoreRoot.vxml" >\n    <meta name="author" content="John O'Reilly"/>\n    <meta name="copyright" content="Sams Publishing 2001"/>\n    <meta name="description" content="Calculate the cost of book order"/>\n    <meta name="keywords" content="VoiceXML Sams Books"/>\n    <meta name="maintainer" content="johnnyo@sams.net"/>\n    <meta name="robots" content="index,follow"/>\n    <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>\n    <meta name="maintainer" content="johnoreilly@lucent.com"/>\n    <meta name="application" content="Book Store"/>\n    <var name="added_cost" />\n    <?php\n        echo '<var name="added_items" expr="';\n        echo $num_of_items;\n        echo '" />';\n    ?>\n    ...
```

12.2.3 数据库访问

应用程序开发人员使用服务器端应用程序的主要原因之一是要从数据库中提取信息，然后用**HTML**对信息进行格式化，以便用浏览器来查看这些信息。我们将要开发一个这种类型的程序例子，然后对它进行修改，使之成为我们的语音应用程序。在这些程序举例中，我们将要说明如何使用配有**Java**数据库连接（**JDBC**）API的**Java**小服务程序和**PHP**。

在你运行这个示例程序之前，你必须安装一个数据库。对于这些示例程序，你将使用 mySQL 数据库。另外，对于 Java/JDBC 示例程序，你需要相应的 JDBC 驱动程序。最后，对于 Java 小服务程序运行环境，你必须对这些工具进行相应的配置，使它们能够一道运行。

PHP 包含了对 mySQL 的支持特性，因此不需要对配置作任何修改。当你完成了 Java 的设置，创建了一些数据以便对它进行测试，并且完成了测试之后，你就可以使用同样的数据来测试你的 PHP/mySQL 配置。

1. 将 mySQL 用于 Java/JDBC

在本小节中，我们将要对你的运行环境进行相应的配置，这样就可以将 mySQL 用于 Java 小服务程序和 JDBC。我们将要编写一个独立的 Java 程序，以便测试这个运行环境，然后转入 Java 小服务程序的操作。

下面的软件是必须的：

- mySQL（从网址 www.mysql.com 下载）。
- MM mySQL JDBC 驱动程序（从网址 <http://mssql.mysql.sourceforge.net> 下载）。

请按照上面列出的软件顺序下载和安装这些软件。安装 mySQL 数据库软件的方法是将下载的软件解压缩到一个临时目录，再运行标准的 setup.exe 安装程序。在安装过程中，系统将只提示你建立安装目录。若要启动数据库运行，请使用命令 <mysqlinstalldir>\bin\mysqld。根据系统提示，启动 mySQL 的 shell 程序 <mysqlinstalldir>\bin\mysql，然后输入下面这个命令，以便建立一个名叫 Books 的新表格，该表格包含的字段有 Title、Price 和 Topics：

```
USE Test;
CREATE table Books (Title varchar(64), Price decimal(3,2), Topics varchar(64));
```

在安装 mySQL 的过程中，将建立不带任何表格的 Test 数据库。你可以检查你建立的 Books 表格是否正确，方法是使用下面的 DESCRIBE 命令：

```
DESCRIBE Books;
```

接着你必须安装 MM mySQL JDBC 驱动程序。若要安装驱动程序，只需要下载它的包含 (jar) 文件，并将它放到你的 Java CLASSPATH 中。首先用一个独立的 Java 应用程序来测试该驱动程序，了解它的安装和配置是否正确。为此你需要将一些抽样数据输入 Books 表格，再编写一个程序来查询该表格的数据。（实际上，该程序可以用来查询任何表格的数据）。

若要将数据填入 Books 表格，请将下列命令输入 mySQL 的 shell 程序：

```
INSERT into Books VALUES ( "Voice Application Development Using VoiceXML",
                            59.95, "VoiceXML, ASR, TTS, Internet" );
INSERT into Books VALUES ( "WAP Development with WML and WMLScript",
                            59.95, "WML, WAP, Internet" );
```

若要查看这些命令插入表格的数据，请使用下面的 SELECT 命令：

```
SELECT * from Books;
```

上面我们介绍了一个简单的独立 JDBC Java 应用程序，它可以用来查询表格，测试你的 JDBC 和你安装的 mySQL 数据库。该程序将能接受各种命令行参数，如主机名、数据库名、用户、口令和 SQL 查询字符串。它将执行查询，并且输出下面的结果：

```
SQLTestApp.java
import java.sql.*;
import java.lang.*;
import java.util.*;

class MySQLTest
{
    public static void main (String args [])
        throws SQLException, ClassNotFoundException
    {
        if (args.length != 5)
        {
            System.out.println("\nUsage: java MySQLTest <host> <database> <login>
                               <passwd> <query>");
            System.exit(1);
        }

        // Load the mySQL JDBC driver
        try
        {
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
        }
        catch( Exception e )
        {
            e.printStackTrace();
        }

        // Connect, query and report
        try
        {
            String DBHost    = args[0];
            String DBDb      = args[1];
            String DBUser    = args[2];
            String DBPasswd = args[3];
            String DBQuery   = args[4];

            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://" + DBHost + "/" + DBDb +
                "?user=" + DBUser + "&password=" + DBPasswd );

            // Create a Statement
            Statement stmt = conn.createStatement ();

            ResultSet rs = stmt.executeQuery ( DBQuery );
            ResultSetMetaData rsmd = rs.getMetaData();
            int numColumns = rsmd.getColumnCount();

            while (rs.next ())
            {
                for( int i = 1; i <= numColumns; i++ )
                {
                    System.out.print( rs.getObject( i ) + "|" );
                }
            }
        }
    }
}
```

```
        }
        System.out.println();
    }
    conn.close();
}
catch( Exception e )
{
    System.out.println( e.getMessage() );
}
}
```

若要编译该程序，请在系统提示符后面输入

```
javac SQLTestApp.java
```

若要执行该程序，请在系统提示符后面输入

```
java SQLTestApp localhost test "" "" "select * from Books"
```

该命令的输出应该是查询的结果，它就是你的Books表格的内容。如果查询失败，请弄清MM JDBC 驱动程序是否在你的CLASSPATH目录中。

现在你已经成功地运行了一个Java JDBC 应用程序。在下一节中，你将要在这个基础上编写一个访问数据库的小服务程序。不过在你着手编写JDBC 小服务程序之前，还要执行一项最后的配置步骤。这就是要使JRun小服务程序引擎能够用于MySQL数据库。

请打开JRun Management Console (JRun管理控制台) (方法是依次点击Start, Programs, JRun, JRun Management Console) 并进行登录。在控制台的左边，是个配置参数的树状视图。请同时在Admin Server和Default Server中选定 Java Settings页，并且修改CLASSPATH，以便添加mySQL包含文件。最后，在Default Server的JDBC数据源下，使用向导来添加 Test 数据库的新数据源。在使用向导时，相关的设置（假定你使用安装的默认值）是：

Name: jdbc:mysql://localhost/Test
URL: jdbc:mysql://127.0.0.1:3306/Test

2. 用PHP测试mySQL

前面讲过，要使 PHP能够用于 mySQL数据库，不需要对配置作任何修改，因为 PHP 软件包含了对mySQL的支持特性。我们可以观察一个简单的示例程序，以便确认你安装的 mySQL能够正确地运行。请将下面的脚本放入你的 Apache Web服务器的 htdocs子目录中。如果要通过Web浏览器来访问该子目录，那么应该显示Books数据库表格的头一行：

```
Data.php
<html>
<body>
<?php
$db = mysql_connect("localhost", "");
mysql_select_db("Test",$db);
$result = mysql_query("SELECT * FROM Books",$db);
printf( "TITLE: %s<br>\n", mysql_result($result,0,"Title"));
printf( "CODE: %s<br>\n", mysql_result($result,0,"Price"));
?>
```

```
</body>
</html>
```

本小节只是介绍如何测试你的mySQL数据库的配置情况，然后再详细介绍如何访问你的语音应用程序中的数据库。在后面的“使用 PHP访问mySQL表格”这一节中，将要介绍如何使用 PHP来访问数据库的一些特定例子。在下一节中，我们要建立一个 JDBC 小服务程序，以便访问你已建立是Books 表格。

3. 用Java小服务程序/JDBC访问mySQL表格

用来说明如何使用 JDBC访问数据库的小服务程序将作为一个小型VoiceXML服务程序来运行。程序清单12-2中的服务程序将能够接受呼叫方讲的语言，并且输出一个 VoiceXML对话框，报告数据库中使用这种语言的符合条件的图书数量及书名。

程序清单12-2 Books.java

```
Books.java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;
import java.util.*;

public class Books extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse resp )
        throws IOException,
               ServletException
    {
        int numMatches = 0;
        Vector matches = new Vector();

        resp.setContentType( "text/xml" );
        PrintWriter out = resp.getWriter();

        // Load the mySQL JDBC driver
        try
        {
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
        }
        catch( Exception e )
        {
            out.println( "Error getting database driver." );
        }

        // Connect, query and report
        try
        {
            String DBHost    = "localhost";
            String DBDb      = "Test";
            String DBUser    = "";
            String DBPass    = "password";
            String query     = "SELECT title, count(*) AS num FROM books WHERE language = ?";

            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://" + DBHost + ":" + DBPort + "/" + DBDb,
                DBUser, DBPass);

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            while( rs.next() )
            {
                numMatches++;
                matches.addElement( rs.getString("title") + " (" + rs.getInt("num") + ")" );
            }
            rs.close();
            stmt.close();
            conn.close();
        }
        catch( Exception e )
        {
            out.println( "Error connecting to database." );
        }
    }
}
```

```
String DBPasswd = "";
String DBQuery = "SELECT Title from Books where Topics LIKE '%" +
    + req.getParameter("keyword") + "%'";

Connection conn = DriverManager.getConnection(
    "jdbc:mysql://" + DBHost + "/" + DBDb +
    "?user=" + DBUser + "&password=" + DBPasswd );

// Create a Statement
Statement stmt = conn.createStatement ();

ResultSet rs = stmt.executeQuery ( DBQuery );

while (rs.next ())
{
    matches.addElement( rs.getString( "Title" ) );
    numMatches++;
}
conn.close();
}
catch( Exception e )
{
    out.println( e.getMessage() );
}
out.println("<?xml version='1.0'?> ");
out.println("<vxml version='1.0'>");

out.println( "<form>" );
if( numMatches == 0 )
{
    out.println( "<block>No books on that subject were found." );
}
else if( numMatches == 1 )
{
    out.println( "<block>We have one book on this subject. It is " );
    out.println( "<break msec="500"/>" );
    out.println( matches.elementAt( 0 ) );
}
else
{
    out.println( "<block>We have " + numMatches + " books on this subject.
➥They are" );
    for( Enumeration e = matches.elements(); e.hasMoreElements(); )
    {
        out.println( e.nextElement() );
        out.println( "<break msec="500"/>" );
    }
}
out.println( "<break msec="500"/>" );
out.println( "</block>" );
out.println( "</form>" );
out.println( "</vxml>" );
}
}
```

由于你还没有办法用电话来为小服务程序提供keyword参数，因此我们可以使用 Web浏览器来测试该小服务程序。将小服务程序的URL输入一个 Web 浏览器，并且输入keyword参数，小服务程序的输出将被返回给浏览器。浏览器不知道如何解释 VoiceXML的含义，但是通过查看返回页面的源代码，你就能够看到该程序的输出。你甚至可以将源代码剪切和粘贴到一个 VSP 的 VoiceXML检验器中，以确定 VoiceXML是否正确。使用某些 VSP的开发工具，你可以呼叫程序开发的电话号码，并且听取其结果。这可以用来检查定时。

输入下面的URL：

```
http://localhost:8100/servlet/Books?keyword=VoiceXML
```

这将使你的小服务程序输出下面的源代码：

```
<?xml version="1.0"?>
<vxm version="1.0">
<form>
<block>
We have one book on this subject. It is
<break msecs="500"/>
Voice Application Development Using VoiceXML
<break msecs="500"/>
</block>
</form>
</vxm>
```

输入下面的URL：

```
http://localhost:8100/servlet/Books?keyword=BellyBoy
```

这将使你的小服务程序输出下面的源代码：

```
<?xml version="1.0"?>
<vxm version="1.0">
<form>
<block>
No books on that subject were found.
<break msecs="500"/>
</block>
</form>
</vxm>
```

我们真正想要做的事情是通过电话来呼叫应用程序，说出一个关键字，让应用程序向我们报告符合关键字条件的图书。为此，我们需要使用一种方法，以便用关键字作为参数来调用小服务程序。因此我们将要编写一个VoiceXML 对话框，向用户查询一个关键字，并且将它传递给 Books.java程序。

为了对此进行测试，你将要建立一个前端 VoiceXML对话框，以便查询该关键字，然后将关键字传递给负责搜索的小服务程序。该对话框包含在下面这个示例程序的 BookSearch.vxml 中：

```
BookSearch.vxml
<?xml version="1.0"?>
<vxm version="1.0">
<form>
<field name="keyword">
```

```

<prompt>
    Please select a web development language <enumerate/>
</prompt>
<option value="HTML"> H T M L </option>
<option value="WML"> W M L </option>
<option value="VoiceXML"> Voice X M L </option>
<filled>
    <submit next="http://www.myisp.net/beez/servlet/Books"
        method="get" namelist="keyword"/>
</filled>
</field>
</form>
</vxml>

```

如果用VSP的开发人员程序来测试上面的示例程序，那么对你运行小服务程序的环境应该提出两个要求：

- 小服务程序必须由支持小服务程序的Web服务器来托管，并且可以通过因特网来访问该Web服务器。
- 小服务程序必须由支持 mySQL的Web服务器托管。

当你将该对话框与负责搜索的小服务程序和数据库一道安装在一个经过恰当配置的主机上后，你就可以配置你的VSP账户，使之指向BookSearch.vxml 程序，并且用电话来测试该搜索工具。

4. 使用PHP访问 mySQL表格

上面我们已经介绍了如何用 Java小服务程序来开发能够访问数据库的 VoiceXML应用程序，下面让我们对PHP的应用作一讲解，与mySQL一样，PHP的使用在ISP中也非常普遍。Unix编程人员对PHP的语法都是熟悉的。如果要编写下面的程序，你只需要对 Books.java程序进行相应的改写即可：

```

Books.php
<?xml version="1.0"?>
<vxml version="1.0">

<form>
<?php
$db = mysql_connect("localhost", "");
mysql_select_db("Test", $db);
$query = "SELECT Title FROM Books WHERE Topics LIKE \"%$keyword%\"";
$result = mysql_query( $query, $db );
$numMatches = mysql_num_rows( $result );
if( $numMatches == 0 )
{
    echo "<block>No books on that subject were found.</block>";
}
elseif( $numMatches == 1 )
{
    echo "<block>We have one book on this subject. It is \n";
    echo mysql_result( $result, 0 );
}

```

```

else
{
    echo "<block>We have $numMatches related books. They are ";
    while( $row = mysql_fetch_array( $result ) )
    {
        echo "\n".$row["Title"]."\\n<pause>500</pause>";
    }
}
echo "\\n<break msecs=\"500\"/>\\n";
echo "</block>\\n";
mysql_free_result ($result);
?>
</form>
</vxmle>

```

你可以像执行Books.java程序那样执行上面这个程序，既可以在浏览器源代码窗口中观察程序运行的结果，也可以从VSP链接到在恰当配置的Web主机上运行的Books.php和BookSearch.vxml程序。

介绍这些程序的目的是告诉你如何用VoiceXML进行服务器端的编程，并且使你能够着手开始进行这样的编程工作。这不一定是用这些工具建立商用应用程序的最佳方法。之所以我们忽略了讲解上是简明扼要，原因之一是试验性应用程序应该层次分明地加以讲解，防止业务逻辑与用户界面紧密纠缠在一起。如果你按照HTML和WAP编程的做法进行工作，那么你就已经拥有一个支持这些界面的数据组件库，现在你可以在此基础上比较容易地建立VoiceXML应用程序。

12.3 加载声音文件

有一个VoiceXML元素，即record标记，几乎必须进行服务器端的编程。虽然你可以非常容易地在一个对话框中进行声音文件的录制和重新播放，也可以将录制的声音文件保存起来供将来播放，但是你必须进行某种形式的服务器端的编程。并不是所有VSP都支持record标记的使用，因此在试图使用record标记之前，请查阅VSP的文档资料。

当使用record标记时，声音将存放在name属性定义的变量中。若要使之保持不变，你必须将该变量提交给你的Web服务器上的一个服务端程序。然后该服务端程序抓取该变量，并且放入一个数据流使之与变量相关的文件中。

首先我们编写一个VoiceXML对话框，它提示呼叫方录制一个消息，再播放该消息，然后，如果呼叫方确认了录制的消息，便调用服务器端的程序，以便保存录制的消息。下面的第一个示例程序说明如何使用Java然后将一个示例放到我们运行的在线书店示例程序之中。程序清单12-3显示了它的前端对话框，它要求用户输入消息，并且调用服务器端的程序来存放该消息。

程序清单12-3 Recording.vxml

```

Recording.vxml
<?xml version="1.0"?>
<vxmle version="1.0">

<form>

```

```

<record name="myaudio" maxtime="8s" finalsilence="3000ms">
  <prompt>Record your message now.</prompt>
  <noinput>I didn't understand. Please try again.</noinput>
</record>

<field name="confirm" type="boolean">
<prompt>Your said <value expr="myaudio" />.</prompt>
<prompt>Would you like to store this message?</prompt>
<filled>
  <if cond="confirm">
    <clear />
  <else />
    <prompt>Now saving your recording.</prompt>
    <submit next="http://www.myisp.net/servlet/Record"
           method="post" namelist="myaudio" />
  </if>
</filled>
</field>
</form>
</vxml>

```

假设呼叫方调用了该应用程序并且进行了下面的对话：

应用程序：现在请录制你的消息。

呼叫方：你应该购买一些关于Belly Boy的书。

应用程序：你说：“你应该购买一些关于 Belly Boy的书”。

应用程序：你想将这条消息存储起来吗？

呼叫方：是的。

应用程序：现在请保存你的录音。

这时，对话框程序 Recording.vxml调用 Record 小服务程序，用变量myaudio 录制声音数据。在我们介绍小服务程序Record.java如何处理声音数据并且将它保存到一个文件中去之前，有一个新工具需要加以介绍，并且要讲述一些配置步骤。

当通过 VoiceXML 提交声音数据时，它是作为HTTP多部分组成的 MIME 窗体数据发送的。声音文件也可以用窗体-url-编码数据来提交，但是，由于这样做比较简单，因此本小节中我们将重点介绍比较复杂的HTTP多部分组成的 MIME窗体数据。处理 Java小服务程序中的输入参数比较简单，我们前面介绍的示例程序已经展示了这一情况。但是 Java API 要支持对多部分组成的数据的处理就不那么容易了，他需要编写相当数量的与你的应用程序没有直接关系的代码，或者购买一个文件上载组件来处理多部分数据的访问请求。通过因特网搜索“Java component multipart file upload”，你可以找到若干这样的文件上载组件。

对于下面这个示例程序，我们将使用Servertec公司（网址是www.servertec.com）提供的文件上载组件。Servertec公司允许你进行90天的免费试用下载，许可证费用也不高。分销的组件包括MultipartForm和FileUpload这两个组件，它们能够帮助完成我们的任务，并且界面也比较简单。首先下载这两个组件并将它们解压缩。请将 <fileuploadinstall dir>/classes添加给你的 CLASSPATH。

现在我们就可以抓取发送给我们的Record小服务程序的声音数据，如程序清单12-4所示。

程序清单12-4 Record.java

```
Record.java
import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import stec.servlet.http.*;

public class Record extends HttpServlet
{
    public void doPost( HttpServletRequest req,
                        HttpServletResponse res )
        throws ServletException, IOException
    {
        ServletOutputStream out = res.getOutputStream();
        String output = "Your message is saved.";
        res.setContentType( "text/xml" );

        try
        {
            ServletContext context = getServletContext();
            String basedir = context.getRealPath( "./audio/" );
            File f = new File( basedir );
            f.mkdirs();

            Hashtable ht = MultiPartForm.parse( req,
                                                f.getCanonicalPath(), 1024*1024 );
        }
        catch( Exception e )
        {
            output = "Your message could not be saved.";
        }
        out.println("<?xml version=\"1.0\"?> ");
        out.println("<vxm version=\"1.0\">");
        out.println("  <form> ");
        out.println("    <block> ");
        out.println("      "+output);
        out.println("    </block> ");
        out.println("  </form> ");
        out.println("</vxm>");
    }
}
```

我们可以像下面的示例程序那样使用PHP来完成同样的操作：

```
Record.php
<?php
echo '<?xml version="1.0"?>';
echo '<vxm version="1.0">';
echo "  <form>";
```

```

echo "<block>";

if (is_uploaded_file($myaudio))
{
    copy($myaudio, "d:\recordings\audiofile");
    echo "Your message is saved";
}
else
{
    echo "Your message could not be saved.";
}

echo "</block>";
echo "</form>";
echo "</vxml>";
?>

```

让我们使用上面这些示例程序来建立实际运行的在线书店示例程序。在我们的示例应用程序中，程序清单12-5中的对话框可以接受来自用户的录音输入。

程序清单12-5 GetComments.vxml

```

GetComments.vxml
<?xml version="1.0"?>
<!--GetComments.vxml-->
<!--This document demonstrates the record command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
<property name="caching" value="safe" />
<meta name="author" content="John O'Reilly"/>
<meta name="copyright" content="Sams Publishing 2001"/>
<meta name="description"
      content="This page collects comments for Sams online bookstore,
      to access call (xxx)-xxx-xxxx"/>
<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
<form id="record_comments">
    <record name="comments" modal="true" beep="true" maxtime="60s"
           finalsilence="2s" dtmfterm="true">
        <prompt>
            Please enter your comments at the beep.
        </prompt>
        <filled>
            <prompt>
                The comments <value expr="comments" mode="recorded"/> have been saved.
            </prompt>
            <submit next="http://www.myisp.net/beez/servlet/RecordComments"
                   method="post" namelist="comments" />
        </filled>
    </record>
</form>
</vxml>

```

该对话框程序调用我们的 Web 服务器上的 RecordComments 小服务程序来保存声音文件。这时程序清单 12-6 所示的 RecordComments 小服务程序将抓取该声音文件，将某个状态报告给用户，然后将控制返回给应用程序中的更高层对话框。

程序清单 12-6 RecordComment.java

```
RecordComments.java
import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import stec.servlet.http.*;

public class RecordComments extends HttpServlet
{
    public void doPost( HttpServletRequest req,
                        HttpServletResponse res )
        throws ServletException, IOException
    {
        ServletOutputStream out = res.getOutputStream();
        String output = "Let's return to the main menu.";
        res.setContentType( "text/xml" );

        try
        {
            ServletContext context = getServletContext();
            String basedir = context.getRealPath( "./audio/" );
            File f = new File( basedir );
            f.mkdirs();
            Hashtable ht = MultiPartForm.parse( req,
                                                f.getCanonicalPath(), 1024*1024 );
        }
        catch( Exception e )
        {
            output = e.toString();
        }

        out.println("<?xml version=\"1.0\"?> " );
        out.println("<!--UploadComments.vxml-->");
        out.println("<vxml version=\"1.0\""
                   " application=\"/BookStoreRoot.vxml\" >");
        out.println("<meta name=\"author\" content=\"Rick Beasley\"/>");
        out.println("<meta name=\"copyright\"");
        out.println("      content=\"Sams Publishing 2001\"/>");
        out.println("<meta name=\"description\"");
        out.println("      content=\"Upload a recorded audio file, \"");
        out.println("              to access call (xxx)-xxx-xxxx\"/>");
        out.println("<meta name=\"keywords\""
                   " content=\"VoiceXML Sams Books\"/>");
        out.println("<meta name=\"maintainer\""
                   " content=\"beez@sams.net\"/>");
        out.println("<meta name=\"robots\" content=\"index,follow\"/>");
```

```

        out.println("<meta http-equiv=\"Date\"");
        out.println(" content=\"Thu, 7 Aug 2000 20:20:20 EST\"/>");
        out.println("<meta name=\"maintainer\" \"");
        out.println(" content=\"johnoreilly@lucent.com\"/>");
        out.println("<meta name=\"application\""
                   content=\"Book Store\"/>");
        out.println(" <form> ");
        out.println(" <block> ");
        out.println(" <+output> ");
        out.println(" <goto"
                   next=\"/http://localhost/WelcomeMenuProp.vxml\"/> ");
        out.println(" </block> ");
        out.println(" </form> ");
        out.println("</vxml> ");
    }
}

```

这时，当用户口述程序注释时，注释将被录音，用户会听到“*The comments comment text have been saved. Let's return to the main menu*”（注释“comment text”已经被保存。让我们返回主菜单），然后将控制返回给 WelcomeMenuProp.vxml 程序，如程序清单 12-7 所示。

程序清单 12-7 WelcomeMenuProp.vxml

```

WelcomeMenuProp.vxml
<?xml version="1.0"?>
<!--WelcomeMenuProp.vxml-->
<!--This document demonstrates the property tag -->
<vxml version="1.0" application="BookStoreRoot.vxml">
<meta name="author" content="John O'Reilly"/>
<meta name="copyright" content="Sams Publishing 2001"/>
<meta name="description"
      content="Welcome page for Sams online bookstore,
      to access call (xxx)-xxx-xxxx"/>
<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
<form id="welcome">
  <block>
    <prompt>
      <audio src="welcome.wav" fetchtimeout="4s" fetchhint="prefetch">
        <pros vol="+20%">
          <div type="paragraph">
            <emp level="strong">Welcome</emp>
            to Sams Publishing's online bookstore.
            Our store allows you to conveniently purchase Voice
            Application Development with Voice XML books and t-shirts.
            You can also visit us on the web at <break msecs="100" />
            <sayas sub="W W W dot voice genie dot com ">
              www.voicegenie.com
            </sayas>

```

```

        </div>
    </pros>
</audio>
To speak to an operator at any time, say operator or press star zero.
Other useful commands are main menu, repeat, and good bye.
</prompt>
<goto next="#what_next" />
</block>
</form>

<menu id="what_next" scope="document" dtmf="true">
<prompt>
    You are at the main menu.
    To add an item or items say add or press 1.
    To remove an item or items say remove or press 2.
    To record comments about the automated book store or any of our products
    say record comments or press 3.
    To list and get the total cost of the items in your shopping cart
    say list or press 4.
    To check out say check out or press 5.
</prompt>
<choice next="AddItemEvent.vxml">add</choice>
<choice next="RemoveItem.vxml">remove</choice>
<choice next="GetComments.vxml">record comments</choice>
<choice next="ListItems.vxml">list</choice>
<choice next="CheckOutScript.vxml">check out</choice>
<catch event="help" cond="global_help">
    You are currently at the bookstore's main menu.
    <assign name="global_help" expr="false" />
    <throw event="help" />
</catch>
</menu>
</vxml>

```

12.4 小结

与HTML一样，VoiceXML主要是个界面开发语言。它是用户与你的应用程序之间的语音界面。虽然VoiceXML和ECMAScript可用于建立动态用户界面，但是不应该将它们与建立你的应用程序的商用逻辑时所用的编程语言相混淆。请使用功能强大的服务器端的编程工具。

服务器端的编程有助于你建立VoiceXML应用程序，它使你能够展示可以通过程序来控制的任何信息。我们介绍了如何访问参数和数据库，作为一个引言，不过从广义上讲，现在我们可以清楚地看到，你能够通过代码来操作的任何信息，你也可以用VoiceXML来展示它们。因此每一种编程语言都具有某些服务器端的脚本编程特点。

我们简单介绍了Java小服务程序和PHP，也讲述了你可能遇到的一些常见编程任务。与理解适合你进行的编程所用的工具类型相比，特定的语言或工具就显得一点儿也不重要了。经过上面两章内容的介绍，我们可以清楚地看到，客户端脚本语言最适合增强用户界面的功能和进行性能的验证，而服务器端脚本语言则有助于完成你的大部分商用逻辑的开发。

第13章 VoiceXML的未来发展前景

最近几年来，将VoiceXML用作迅速开发和部署具备 Web访问功能的电话服务程序的语言，已经成为一股势不可挡的潮流。使用Nuance、Tellme 和VoiceGenie开发网络的开发人员数量不断增长，就说明了这样一个发展趋势。许多重要的企业组建了 VoiceXML 论坛，并且部署了语音门户来支持以语音应用为基础的电话市场，进一步说明了这个问题。

自从VoiceXML论坛开发出 VoiceXML 1.0 技术标准以来，接着又通过它的正式推荐程序推出了VoiceXML 2.0技术标准，由此万维网联盟（W3C）出现了许多变化。它的分阶段实施方案包括一系列工作草案文档，接着是最后定稿的工作草案，候选的建议书，推荐的建议书，以及最后定稿的建议书。VoiceXML 2.0技术标准的目的是澄清 VoiceXML 1.0 技术标准的若干领域存在的问题，增加和修改一些元素和属性，并且就基于XML的 Web应用程序环境中的语法和语音合成的使用标准提出了建议。

但是，由于我们的公司是 VoiceXML 论坛和W3C的语音浏览器集团的成员，并且由于成员资格带来的知识产权问题，我们不能随意泄露 VoiceXML 技术标准目前的细节。等到2001年夏季的某个时候，当该技术标准的工作草案发布以征求公众意见的时候，这种限制就会解除。本书的作者打算在以后更加详细地介绍VoiceXML 2.0技术标准。与此同时，本章将重点介绍“用于语音界面框架的语音合成标记语言技术规范”和“用于 W3C 语音界面框架的语音识别语法技术规范”，因为公众将从2001年1月3日起可以获得的这些文档的工作草案。

13.1 关于语音合成标记语言的概述

语音合成标记语言（SSML）技术规范是在Java语音标记语言（JSML）的基础上开发而成的，它被定义为一个XML应用语言。SSML 定义了一组元素和属性，语音合成器应该使用它们来再现由SSML 或VoiceXML 文档产生的语音，以前，用于VoiceXML的语音标准依赖于 SABLE 联盟和 JSML 的技术规范。VoiceXML 1.0技术规范将许多语音标记的定义交给解释程序的实现人员去确定了。他有时必须从前面提到的两个技术标准中获取有关的特性，或者对标记语言进行扩展，以便将它映射为基础语音合成器能够处理的东西。显然，这样的安排将会妨碍人们采用 VoiceXML 。SSML技术标准是一种尝试，目的是对 VoiceXML 应用程序应该适应的界面实施标准化，以便支持最大的可移植性。表13-1汇总列出了VoiceXML 1.0与2.0 技术标准使用的标记语言之间的部分主要差异。

表13-1 VoiceXML1.0与2.0的比较

VoiceXML 2.0的元素	VoiceXML 2.0的属性	对VoiceXML 1.0的修改
speak	xml: lang	Speak是SSML的新元素，不过在VoiceXML 2.0 中通常不用

(续)

VoiceXML 2.0的元素	VoiceXML 2.0的属性	对VoiceXML 1.0的修改
paragraph		相当于VoiceXML 1.0中<vxm1>元素的language属性
sentence		原先是VoiceXML 1.0 中的<dir>的属性值
say-as	type	原先是VoiceXML 1.0中的 <dir>的属性值 在 VoiceXML 2.0中 增加了 duration、measure、name、net和 address 等新类；增加了对date的支持
phoneme	alphabet ph	原先是VoiceXML 1.0的 <sayas>属性值
voice	gender age category variant name	<voice>是 VoiceXML 2.0的新属性
emphasis	level	用于取代VoiceXML 1.0中的<emp>元素，不过它的含义相同
break	size time	Time取代VoiceXML 1.0中的msecs
prosody	pitch contour range rate duration volume	Counter和duration是VoiceXML 2.0中的新属性；<prosody>和 volume 分别取代<pros>和 vol
audio	src	在VoiceXML 2.0中新建立的
mark	name	在VoiceXML 2.0中新建立的

13.2 关于语音识别语法的概述

语音识别语法技术规范是VoiceXML 2.0文档必须使用的技术规范。语法技术规范是由专用Jspeech语法格式、PipeBeach语法、TalkML和 Jspeech语法格式的一种研究变形中发现的语言特性派生而来的。语法技术规范本身由两种句法格式组成，一种格式是强化 BNF (ABNF) 句法，另一种是XML 句法。ABNF是一种明文格式，它类似常见的 BNF 格式。BNF 与它的各种变形构成了语音软件供应商提供的专用语法规范语言 (GSL) 的基础。基于 XML 的语音识别语法格式使用XML元素和属性来表示语法。该技术规范还要求这些格式之间能够互相转换。

语音识别语法技术规范包含了许多增强特性，这是VoiceXML 1.0 支持的语法技术规范所没有的。下面是其中的一些增强特性：

- 权标——用于定义口语说出的合法词汇，它带有可能返回的成功结果。权标可以用双引号(“ ”)、空格或<token> 元素标记作为限定符。

- 规则引用——规则用于定义语法中可以说出的东西，而规则名则用于定义语法作用域中这些规则的独一无二的标识。语法规范用于定义对这些规则的若干类型的引用，包括本地引用和URL的外部引用。
- 词序——这是符合规则的一个扩展词序，用于定义用户必须说的和语音识别程序必须检测到的扩展词汇的临时次序。例如，假设一个词序由“this apple is \$ taste”定义，其中 \$ taste定义为(sour|sweet)。扩展词序“this apple is red”虽然符合上面的词序，但是它是无效词序，因为red不是\$taste的有效值。
- 替代扩展词串——使用符合规则的扩展词串，可以定义一列由竖线“|”隔开的替代选择词汇。用于设定每个选择的可能性的加权值是视情况而定的。
- 标记——标记是视情况而定的字符串，附加在符合规则的扩展词串的后面，帮助发送语音识别结果的处理。标记对语音识别本身没有任何影响。

表13-1和上面列出的增强特性并不能全面展示VoiceXML 2.0所需要的新语言和语法规范。为此，我们希望你访问<http://www.w3c.org>，以便了解关于这些技术标准演变发展的详细信息。

13.3 多模式应用程序

VoiceXML基本上属于一种单一模式语言。无论你是说话还是使用DTMF小键盘，它的工作模式都是“声音进，声音出”。单一的输入/输出模式的局限性使得VoiceXML应用程序具备了可以在任何时间和任何地点通过电话来访问应用程序的特点。这种局限性也限制了VoiceXML应用程序能够处理的数据类型。此外，当前的VoiceXML应用程序使得失明的用户也能够访问因特网，但是有语言或听力缺陷的用户则无法通过VoiceXML来访问因特网。显然，有必要将各种类型的浏览器功能组合、结合或协调起来，使得大多数人都能够最有效地实现因特网的各种应用。

看来将来的VoiceXML将包含一个多模式应用的组件，从而对用户界面和应用程序的设计产生巨大的影响。表13-2列出了可以在将来的多模应用中组合起来的输入和输出模式。

表13-2 多模式应用程序中可能使用的输入/输出模式

类 别	输入或输出	举 例
声音	输入	用户的语言输入 按DTMF键
	输出	合成语音 录制的声音或非口语语音 流式语音
图像	输出	移动电话、PDA、手持设备或个人计算机上的显示屏
小键盘或键盘	输入	电话小键盘（当用于输入数据时便可使用，比如WML应用中的情况那样） PDA、手持设备或个人计算机上的键盘
定向设备	输入	鼠标、触摸屏、眼睛移动跟踪设备
其他	输入/输出	通常与游戏相关的各种设备，比如特定应用的控制设备和反馈设备（能够振动、发出味道或迫使身体移动的设备）

表13-2并没有包括所有的输入、输出模式，不过由此你可以得到某些概念，了解VoiceXML将会如何成为将来的多模式应用程序。

13.3.1 多模式应用程序给VUI带来的影响

为了更好地理解多模式应用程序给VoiceXML用户界面（VUI）和应用程序的设计带来的影响，我们将要复习一下这种应用程序可能具备的特点，并且要介绍一下它们对VUI的设计会产生哪些影响。多模应用程序将会具备与输入和输出相关的非常重要的特点。

与输入模式的活动相关的输入方式包括下列几种（按复杂性的增序排列）：

- 串行输入——某个时间内只能有一种输入模式处于有效状态。
- 非协调式同时输入——同时可以有多种输入模式处于有效状态，但它们的输入的处理是相互分开的。
- 协调式同时输入——同时可以有多种输入模式处于有效状态，但是它们的输入的处理是互相关联的。

与多种模式的输出如何同步相关的输出方式包括（按复杂性的增序排列）：

- 串行输出——可以使用多种输出介质，不过每次只有一种输出介质可以传递给用户。
- 非协调式同时输出——在一次会话期间可以输出多种介质，但这些介质互不同步。
- 协调式输出或同步输出——同时可以输出多种介质，这些输出是互相协调和同步的。协调与同步之间的差别在于，前者是确保语音应用程序说出的词语与屏幕上显示词语相同，而后者则是确保口语输出的形式与口形模仿相同步。

13.3.2 串行输入/输出

串行输入/输出模式是指在某个时间内只有一种输入或输出模式处于有效状态的多模应用。这是最简的多模应用形式。

使用串行输入/输出模式的应用可以在不同输入设备之间进行转换，比如从识别用户语音转换为识别来自触摸屏的输入，然后又回到识别用户的语音。

表13-3说明了各种串行输入/输出的情况。

表13-3 串行输入/输出的举例

操作步骤	执行的操作	输入/输出模式
用户连接到VoiceXML 应用程序	应用程序作出应答说：“你要购 买多少本VoiceXML图书？”	输入：无 输出：语音
用户作出应答：“1”	应用程序作出应答说：“用电话 键盘输入你的信用卡号码。”	输入：语音（忽略其他输入设备
	这时电话显示屏上出现一个窗体 字段	输出：语音和小型屏幕用户界面
用户输入并且提交信用 卡号码；屏幕显示消失	应用程序作出应答：“谢谢你购 买图书。”	输入：小键盘（忽略语音输入） 输出：语音

为了进行比较,请注意语音输入/输出可以通过配有话筒和扬声器的PC声卡来进行。小键盘输入可以通过PC键盘、鼠标、按压触摸屏等方式来进行。

13.3.3 非协调式同时输入/输出

非协调式同时输入/输出模式是指某个时间可以同时有多个输入或输出模式处于有效状态的多模式应用。对于任何一个输入事件来说,这种输入模式只能接受一种输入,比如将信息输入一个窗体字段。这种输出模式可以同时将多个介质传递给不同的设备,但是这些输出只能协调为它们与同一个对话框相关的程度。

例如,用户可以通过移动电话来访问应用程序,并且接收到一个语音登录提示和一个小型屏幕用户界面的登录对话框。用户如果要进行登录,可以说出他的登录名,也可以通过电话小键盘输入登录名。当用户用一种输入设备打开登录对话框时,其他输入方式将被忽略,直到下一个输入事件为止。

表13-4给出了非协调式同时输入/输出方式的举例。

表13-4 非协调式同时输入/输出方式举例

操作步骤	执行的步骤	输入/输出模式
用户通过PC连接到VoiceXML应用程序	PC上打开一个可以访问Web的用户界面窗口,显示一个登录名和口令的对话框。应用程序应答说:“What is your login name?”(请问你的登录名是什么?)	输入:无 输出:语音,通过Web用户界面显示的图像
用户作出应答说:“Charles one oh five。”	应用程序作出应答:“What is your password?”(你的口令是什么?)	输入:语音(忽略Web用户界面,直到登录事件完成之后) 输出:语音

13.3.4 协调式同时输入/输出

协调式同时输入/输出是指某个时间可以同时有多种输入方式处于有效状态,并且同时有多种输出方式处于有效状态的多模式应用。它的输入模式能够接受任何一个输入事件的扩展输入,并且用一种智能方法实现各种输入的协调。它的输出模式能够提供多种输出,并且这些输出之间能够做到紧密协调。

例如,用户可以使用协调式同时输入输出模式访问自动取款机。当用户触摸自动取款机上的“Saving Account”(储蓄账户)标签时,他可以说:“Withdraw one hundred dollars”(取款100美元)。应用程序将对用户的输入进行识别,认为用户想从他的储蓄账户中取款100美元。

表13-5说明了一个协调式同时输入/输出的例子。

表13-5 协调式同时输入/输出的例子

操作步骤	执行的操作	输入/输出模式
用户走到媒体商店里的音乐搜索台前	音乐搜索台的触摸屏上显示出音乐种类的图标	输入: 用户输入使用的位置

(续)

操作步骤	执行的操作	输入/输出模式
在VoiceXML提示结束前，用户触模屏幕上的“Blues”图标	应用程序感觉到用户对触模屏的触摸，于是说：“What category of music are you interested in？”（你喜欢何种音乐？） 音乐搜索台的触模屏开始显示按字母顺序排列的布鲁斯歌曲演唱者的名单 应用程序问：“What artist？”（你要那位演唱者的歌曲？）	输出：语音，通过触模屏显示的图形 输入：触模屏 输出：语音和图形用户界面
在音乐搜索台的图形用户界面完成前10位演唱者名字的显示之前，用户回答说：“Sonny Boy Willianmson”	音乐搜索台的触模屏开始显示按字母顺序排列的Sonny Boy Williamson 演唱的歌曲名字的列表 语音应用程序问：“Whice title”（哪首歌曲？）	输入：语音 输出：语音和图书用户界面
用户说：“That's the one”（就是这一首），然后触摸“Bye Bye Bird”这首歌的图标	音乐搜索台的屏幕显示关于“Bye Bye Bird”这首歌的信息 语音应用程序回答说：“Do you want to purchase ‘Bye Bye Bird’？”（你要购买Bye Bye Bird这首歌吗？）	输入：触模屏 输出：语音和图书用户界面 忽略“That's the one”这句话，因为这个输入无法处理 输出：语音和图书用户界面

13.4 未来用户界面的复杂性

由于多模式应用程序的功能与未来的VoiceXML相关联，因此语音用户界面和其他用户界面组件将变得越来越复杂。图13-1显示了随着多模功能添加给VoiceXML应用程序，VoiceXML应用程序的复杂性不断提高的趋势。

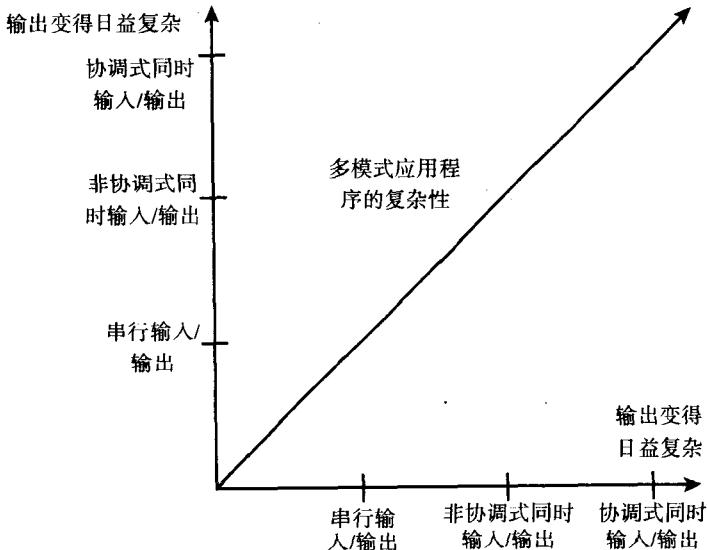


图13-1 多模式功能将会提高未来VoiceXML应用程序的复杂性

在应用程序的设计和开发阶段解决未来应用程序的复杂性问题

随着应用程序的输入和输出选项变得日益复杂，软件应用程序的设计人员、工程师和开发人员应该如何解决这些问题呢？使用第5章介绍的以用户为中心的应用程序设计过程将有助于解决这个问题。

由于信息技术的复杂性越来越大，因此通过用户界面与用户进行通信的复杂性也会随之而增加。由于多模式应用程序配有一个或多个用户界面，因此用户更加容易将它们混淆起来，搞得心烦意乱，甚至怒气冲天。我想将来的VoiceXML应用程序面临的最大挑战是要明智地使用多模式功能，使应用程序的操作更加直观。

以用户为中心的应用程序设计（UCD）有可能成为未来VoiceXML应用程序成败的关键。在多模应用程序开发期间，如果运用UCD过程，那么就能够：

- 确定如何使用多模输入/输出功能来提高用户执行操作任务的效率。
- 确定对目标用户群体最具吸引力的输入方法和输出媒体。
- 测试多模交互操作方式，确保它们适用于最终用户。

未来的VoiceXML应用程序如果与多模基础设施相结合，将能带来许多挑战和机遇。

13.5 VoiceXML多模应用程序的现状

到撰写本书时为止，W3C的“语音标记语言的多模式技术规范”是目前来说最早的一个技术文档草案。此后，它要经过若干个阶段，才能成为工作草案。与此同时，仍然可以“模拟”某些语音和数据的多模交互操作方式。这些交互操作方式中最简单的将是使用VoiceXML来处理所有的语音交互操作和使用WML来处理数据交互操作。

例如，假设一个用户有一部WAP电话，他正在与一个在线电话目录服务进行交互操作。该用户按键输入一个查询字符串，并且将它发送给拥有目录服务的内容提供商。该内容提供商向用户提供了一个符合搜索条件的项目列表。这时用户选定了他需要的项目，但是他希望该项服务能够将用户连接到对应的电话号码。由于该用户的WAP电话只具备单模功能，也就是说它在同一时间内只能支持数据（WAP）或语音输入/输出，而不能同时支持数据和语音输入/输出，因此必须放弃数据会话才能进行出局语音呼叫。另外，值得注意的是有些供应商正在考虑将语音识别功能纳入无线电话设备。使用这种技术，就能够在WAP使用的同一个数据信道上抓取关键字，然后对它进行编码和发送。

13.6 小结

服务提供商和开发商将通过采用VoiceXML 2.0技术标准来保证将来VoiceXML沿着正确的道路发展，并且使VoiceXML能够被人们更加广泛地接受。VoiceXML 2.0不仅消除了VoiceXML 1.0技术标准中存在的定义模糊和歧义等问题，而且通过采用语音合成和语音语法识别格式等手段，防止出现更多的定义模糊问题。多模式技术给VoiceXML应用和发展开拓了新的领域。但是，在多模式技术规范成为工作草案之前，开发人员必须创造性地将VoiceXML、WML和SMS等技术综合起来，以便取得类似的效果。

附录A VoiceXML 1.0的元素

表A-1列出了VoiceXML 1.0的元素。该表对每个元素进行了说明，并且列出了是本书中的哪一章对它进行了介绍。

表A-1 VoiceXML 1.0的元素和它出现在哪一章

元 素	说 明	章号
assign	将一个值赋予一个变量	8
audio	向用户播发一个声音文件	9
block	将可执行代码括起来	8
break	将一个停顿插入语音输出	9
catch	包含一个特定事件的处理程序	9
choice	包含一个菜单选项	9
clear	将值Undefined赋予一个或多个窗体项变量，并且将事件计数器复置为0	8
disconnect	中止对VoiceXML平台的电话呼叫	8
div	确定一段话是作为一个句子还是一个段落来说出	9
dtmf	设定一个只包含按键拨号的DTMF项的语法	9
else	用于条件语句	8
elseif	用于条件语句	8
emp	用于控制口述的一段文字的强调语气	9
enumerate	输出菜单中的选项，或者输出选项列表中的选项	9
error	包含一个事件处理程序，用于处理应用程序或平台的错误	9
exit	用于结束用户的VoiceXML会话	9
field	用于接收用户的输入	9
filled	当它的窗体项变量被赋予值时便执行	8
form	用于定义一个对话框	8
goto	用于转至另一个对话框或项目	8
grammar	用于设定一个语音语法	9
help	包含一个用户请求帮助时使用的事件处理程序	9
if	用于条件逻辑	8
initial	用于设定进入混合式初始窗体之前的初始处理指令	9
link	用于设定链接的语法被识别时浏览到的位置或产生的事件	9
menu	一个对话框，它的每个菜单都包含语法项和产生的浏览位置	9
meta	用于设定关于VoiceXML文档的信息	8
noinput	包含用户不进行应答时的事件句柄	9
nomatch	包含用户的应答与有效的语法不一致时的事件句柄	9
object	与特定平台的扩展特性之间的一个界面	10

(续)

元素	说 明	章号
option	用于定义一个字段选项，它可以取代语法的提供	9
param	用于设定一个传递给对象或子对话框的参数	8
prompt	将文语转换(TTS)和声音文件的输出括起来	9
property	用于设定应用程序的属性	10
pros	用于确定语音输出的语调特征(如音高，音高范围，速度和音量)	9
record	录制用户说的话，并且将它存储起来，以便以后在同一个对话框中播放	10
reprompt	重新播放上一个提示	9
return	从子对话框返回	8
sayas	确定词组如何发音	9
script	将客户端ECMAScript脚本括起来	10
subdialog	一个VoiceXML函数	8
submit	将参数和程序控制传递给服务器端的脚本，该脚本将生产下一个要执行的VoiceXML文档	8
throw	触发一个事件	9
transfer	将用户与另一方连接起来	10
value	输出一个表达式的值	8
var	声明一个变量	8
vxml	将所有Vxml标记括起来	8

附录B 在线书店应用程序

下面的代码是本书中的许多示例程序使用的完整的在线书店应用程序：

```
<?xml version="1.0"?>
<!--Root document -->
<vxm version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>
    <meta name="description"
          content="Root document for our book store,
                  to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <!-- properties -->
    <property name="fetchaudio" value="fetchaudio.wav"/>

    <!-- variables -->
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />
    <var name="has_coupon" expr="false" />
    <var name="asked_about_coupon" expr="false" />
    <var name="unit_book_price" expr="24.95" />
    <var name="unit_tshirt_price" expr="19.95" />
    <var name="current_item" expr="'book'"/>
    <var name="global_help" expr="true"/>

    <!-- links -->
    <link next="CallOperator.vxml" fetchaudio="fetchaudio.wav">
        <grammar type="application/x-jsgf"> operator </grammar>
        <dtmf type="application/x-jsgf"> "*" 0 </dtmf>
    </link>

    <link next="WelcomeMenuProp.vxml#what_next"
          fetchaudio="fetchaudio.wav">
        <grammar type="application/x-jsgf">
            main menu | start [over] | home
        </grammar>
    </link>
    <link event="help">
```

```
<grammar type="application/x-jsgf"> help </grammar>
</link>

<link event="exit">
    <grammar type="application/x-jsgf"> [good] bye </grammar>
</link>

<link event="online_bookstore.repeat">
    <grammar type="application/x-jsgf"> repeat </grammar>
</link>

<!-- event handlers -->
<catch event="noinput" count="1">
    I'm sorry I could not hear you. Let's try again.
    <reprompt/>
</catch>

<catch event="noinput" count="2">
    I'm sorry I still could not hear you.
    I am transfering you to an operator right now.
    <goto next="CallOperator.vxml" " />
</catch>

<catch event="nomatch" count="1">
    I'm sorry I didn't understand that.
    Let's try again.
    <reprompt/>
</catch>

<catch event="nomatch" count="2">
    I'm sorry I didn't understand that.
    I am transfering you to an operator right now.
    <goto next="CallOperator.vxml" " />
</catch>

<catch event="exit">
    Thank you for shopping at Sams Online Bookstore.
    Have a nice day!
    <exit/>
</catch>

<catch event="help">
    <assign name="global_help" expr="true" />
    To speak to an operator at any time,
    say operator or press star zero.
    To return to the main menu.
    To listen to a prompt again say repeat.
    To end this call say good bye.
    <reprompt/>
</catch>

<catch event="error.online_bookstore.negative_shopping_cart_value">
```

```
I'm sorry. Our system is experiencing technical difficulties.  
Please call back at a later time.  
    <throw event="exit" />  
  </catch>  
  
  <catch event="online_bookstore.repeat">  
    <reprompt/>  
  </catch>  
</vxml>  
  
<?xml version="1.0"?>  
<!--CallOperator.vxml-->  
<!--This document demonstrates the transfer command -->  
<vxml version="1.0" application="BookStoreRoot.vxml">  
  <meta name="author" content="John O'Reilly"/>  
  <meta name="copyright" content="Sams Publishing 2001"/>  
  <meta name="description"  
        content="This page transfers the user to an operator,  
        to access call (xxx)-xxx-xxxx"/>  
  <meta name="keywords" content="VoiceXML Sams Books"/>  
  <meta name="maintainer" content="johnnnyo@sams.net"/>  
  <meta name="robots" content="index,follow"/>  
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>  
  <form>  
    <transfer name="call" dest="phone://9195558965" bridge="false"  
             connecttimeout="8s" >  
      <prompt>You are being transferred to an operator</prompt>  
      <filled>  
        <if cond="call == noanswer">  
          <prompt>  
            We're sorry, no operators are available at this time.  
            Please try calling again later. Good-bye.  
          </prompt>  
        </if>  
      </filled>  
    </transfer>  
  </form>  
</vxml>  
<?xml version="1.0"?>  
<!--WelcomeMenuProp.vxml-->  
<!--This document demonstrates the property tag -->  
<vxml version="1.0" application="BookStoreRoot.vxml">  
  <meta name="author" content="John O'Reilly"/>  
  <meta name="copyright" content="Sams Publishing 2001"/>  
  <meta name="description"  
        content="Welcome page for Sams online bookstore,  
        to access call (xxx)-xxx-xxxx"/>  
  <meta name="keywords" content="VoiceXML Sams Books"/>  
  <meta name="maintainer" content="johnnnyo@sams.net"/>  
  <meta name="robots" content="index,follow"/>  
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>  
  <form id="welcome">
```

```

<block>
  <prompt>
    <audio src="welcome.wav" fetchtimeout="4s" fetchhint="prefetch">
      <pros vol="+20%">
        <div type="paragraph">
          <emp level="strong">Welcome</emp>
          to Sams Publishing's online bookstore.
          Our store allows you to conveniently purchase Voice
          Application Development with Voice XML books and t-shirts.
          You can also visit us on the web at <break msecs="100" />
          <sayas sub="W W W dot voice genie dot com ">
            www.VoiceGenie.com
          </sayas>
        </div>
      </pros>
    </audio>
    To speak to an operator at any time, say operator or press star zero.
    Other useful commands are main menu, repeat, and good bye.
  </prompt>
  <goto next="#what_next" />
</block>
</form>

<menu id="what_next" scope="document" dtmf="true">
  <prompt>
    You are at the main menu.
    To add an item or items say add or press 1.
    To remove an item or items say remove or press 2.
    To record comments about the automated book store or any of our products
    say record comments or press 3.
    To list and get the total cost of the items in your shopping cart
    say list or press 4.
    To check out say check out or press 5.
  </prompt>
  <choice next="AddItemEvent.vxml">add</choice>
  <choice next="RemoveItem.vxml">remove</choice>
  <choice next="GetComments.vxml">record comments</choice>
  <choice next="ListItems.vxml">list</choice>
  <choice next="CheckOutScript.vxml">check out</choice>
  <catch event="help" cond="global_help">
    You are currently at the bookstore's main menu.
    <assign name="global_help" expr="false" />
    <throw event="help" />
  </catch>
</menu>
</vxml>

<?xml version="1.0"?>
<!--AddItemEvent.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" />

```

```
        content="Sams Publishing 2001" />
<meta name="description"
      content="Select Item to add, to access call (xxx)-xxx-xxxx" />
<meta name="keywords" content="VoiceXML Sams Books" />
<meta name="maintainer" content="johnnyo@sams.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date"
      content="Thu, 7 Aug 2000 20:20:20 EST" />
<meta name="maintainer" content="johnoreilly@lucent.com" />
<meta name="application" content="Book Store" />
<property name="caching" value="safe" />

<form id="pick_item">
    <field name="item">
        <prompt>
            <enumerate>
                To add <value expr="_prompt"/> to your shopping cart,
                say <value expr="_prompt"/> or press
                <value expr="_dtmf"/>.
            </enumerate>
        </prompt>
        <option dtmf="1" value="book"> books </option>
        <option dtmf="2" value="shirt"> tee shirts </option>
        <filled>
            <assign name="application.current_item" expr="item" />
            <goto expr="(item=='book' &&
                        application.asked_about_coupon==false)?
                        'GetCouponInfoEvent.vxml' :
                        'PurchaseStuffSubmit.vxml'" />
        </filled>
        <help cond="global_help">
            You are currently selecting an item to add to your shopping
            cart. Say the name of, or press the phone key corresponding
            to the item you would like to add.
            <assign name="global_help" expr="false" />
            <throw event="help" />
        </help>
    </field>
</form>
</vxml>

<?xml version="1.0"?>
<!--GetCouponInfoEvent.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
    <meta name="author" content="John O'Reilly" />
    <meta name="copyright"
          content="Sams Publishing 2001" />
    <meta name="description"
          content="Get coupon from user if he or she has one,
          to access call (xxx)-xxx-xxxx" />
    <meta name="keywords" content="VoiceXML Sams Books" />
    <meta name="maintainer" content="johnnyo@sams.net" />
```

```
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date"
      content="Thu, 7 Aug 2000 20:20:20 EST"/>
<meta name="maintainer" content="johnoreilly@lucent.com"/>
<meta name="application" content="Book Store"/>
<property name="caching" value="safe" />

<form id="ask_for_coupon">
    <field name="whether_coupon" type="boolean?y=1;n=0">
        <prompt>
            Do you have a coupon?
        </prompt>
        <help cond="global_help">
            If you have a coupon, say yes or press the one key on your
            phone. If you do not have a coupon, say no or press the
            zero key on your phone.

            <assign name="global_help" expr="false" />
            <throw event="help" />
        </help>
        <filled>
            <assign name="application.asked_about_coupon" expr="true"/>
            <if cond="whether_coupon==true" >
                <goto nextitem="coupon_number" />
            <else/>
                <!-- if the user does not have a coupon,
                    just transition to the add x items to the shopping
                    cart page. -->
                <goto next="PurchaseStuffSubmit.vxml" />
            </if>
        </filled>
    </field>

    <field name="coupon_number" type="digits?length=5" >
        <prompt>
            Please enter your five digit coupon number?
        </prompt>
        <help cond="global_help">
            The coupon number should be located at the bottom of the
            email you received informing you of this special offer.
            Please either say or enter the number using the number keys
            on your phone.
            <assign name="global_help" expr="false" />
            <throw event="help" />
        </help>
        <filled>
            <!-- For now, all valid coupons will have the same number -->
            <if cond="coupon_number=='55555'">
                <assign name="application.has_coupon" expr="true" />
                <goto next="PurchaseStuffEvent.vxml" />
            <else/>
                <!-- if the user does not want to enter another coupon

```

```
number, just transition to the add x items to
the shopping cart page. -->
<goto nextitem="try_again" />
</if>
</filled>
</field>

<field name="try_again" type="boolean?y=1;n=0">
<prompt count="1">
    <value expr="coupon_number" class="digits" />
    is not a valid coupon number. Would you like to try again?
</prompt>
<prompt count="2">
    Would you like to try again?
</prompt>

<help cond="global_help">
    If you would like to enter your coupon number again,
    say yes or press the one key on your phone.
    If you would like to proceed with adding books to your
    shopping cart, say no or press the zero key on your phone.

    <assign name="global_help" expr="false" />
    <throw event="help" />
</help>
<filled>
<if cond="try_again==true">
    <clear namelist="coupon_number try_again"/>
    <goto nextitem="coupon_number" />
<else/>
    <!-- if the user does not want to enter another coupon
        number, just transition to the add x items to
        shopping cart page. -->

    <clear namelist="whether_coupon coupon_number
                try_again"/>
    <goto next="PurchaseStuffSubmit.vxml" />
</if>
</filled>
</field>
</form>
</vxml>

<?xml version="1.0"?>
<!--PurchaseStuffSubmit.vxml-->
<!--This document demonstrates the submit element -->
<vxml version="1.0" application="BookStoreRoot.vxml">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
          content="Add j items to a shopping cart,
          to access call (xxx)-xxx-xxxx"/>
```

```

<meta name="keywords" content="VoiceXML Sams Books" />
<meta name="maintainer" content="johnnyo@sams.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />
<form id="collect_items">
    <property name="bargein" value="true" />
    <property name="timeout" value="4s" />
    <catch event="error.invalid_quantity">
        I'm sorry, <value expr="num_of_items" /> is not a valid entry.
        Order quantities can not be fractions or negative.
        Let's try again.
        <clear namelist="num_of_items confirm" />
        <goto nextitem="num_of_items" />
    </catch>

    <field name="num_of_items" type="number">
        <prompt>
            How many <value expr="application.current_item + 's'" />
            would you like to order?
        </prompt>
        <catch event="help" cond="global_help">
            Say or enter using the number keys on your phone the quantity
            of <value expr="application.current_item + 's'" />
            you wish to add to your shopping cart.

            <assign name="global_help" expr="false" />
            <throw event="help" />
        </catch>
    </field>

    <field name="confirm" type="boolean?y=1;n=0">
        <prompt>
            You have ordered
            <value expr="num_of_items" /> <!-- number of items -->
            <!-- singular or plural of the item being added -->
            <value expr="application.current_item +
                (num_of_items != 1 ? 's' : '')" />

            Is this correct?
        </prompt>
        <catch event="help" cond="global_help">
            If the quantity you entered is correct, say yes or press the
            one key on your phone. If you would like to reenter the
            quantity of <value expr="application.current_item +
                (num_of_items != 1 ? 's' : '')" />
            to add, say no or press the zero key on your phone.
        <assign name="global_help" expr="false" />
        <throw event="help" />
    </catch>

```

```

</field>
<filled namelist="num_of_items confirm">
    <!-- Only whole positive quantities will be be excepted -->
    <!-- Make sure the user entered an integer -->
    <var name="is_fraction"
        expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
    <var name="is_negative" expr="num_of_items < 0" />

    <if cond="confirm == false">
        <clear namelist="num_of_items confirm" />
    <elseif cond="num_of_items == 0"/>
        <prompt>
            You have chosen not to add any
            <value expr="application.current_item + 's'" />
            to your shopping cart.
        </prompt>
        <goto next="WelcomeMenuProp.vxml#what_next" />
    <elseif cond="is_fraction || is_negative" />
        <throw event="error.invalid_quantity" />
    <else/>
        <if cond="application.current_item=='book'" >
            <assign name="application.total_books"
                expr="application.total_books
                    + Number(num_of_items)" />
        <else/>
            <assign name="application.total_shirts"
                expr="application.total_shirts
                    + Number(num_of_items)" />
        </if>
        <submit next=".//servlets/CalculateCostAdd"
            namelist="num_of_items" fetchtimeout="3s" />
    </if>
    </filled>
</form>
</vxml>

import javax.servlet.*;
import javax.servlet.http.*;
import java.text.*;
import java.io.*;
import java.net.*;

public class CalculateCostAdd extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse res )
        throws ServletException, IOException
    {
        res.setContentType( "text/xml" );
        ServletOutputStream out = res.getOutputStream();

        String num_of_items = req.getParameter("num_of_items");

        out.println("<?xml version='1.0'?"> " );

```

```

out.println("<!--CalculateCostAdd.vxml-->");
out.println("<vxml version=\"1.0\" \"");
out.println("    application=\"BookStoreRoot.vxml\"");
out.println("<meta name=\"author\" content=\"John O'Reilly\"/>");
out.println("<meta name=\"copyright\"\"");
out.println("content=\"Sams Publishing 2001\"/>");
out.println("<meta name=\"description\"\"");
out.println("content=\"Calculate the cost of a book order, ");
out.println("        to access call (xxx)-xxx-xxxx\"/>");
out.println("<meta name=\"keywords\" content=\"VoiceXML Sams Books\"/>");
out.println("<meta name=\"maintainer\" content=\"johnnyo@sams.net\"/>");
out.println("<meta name=\"robots\" content=\"index,follow\"/>");
out.println("<meta http-equiv=\"Date\"");
out.println("        content=\"Thu, 7 Aug 2000 20:20:20 EST\"/>");
out.println("<meta name=\"maintainer\"\"");
out.println("        content=\"johnoreilly@lucent.com\"/>");
out.println("<meta name=\"application\" content=\"Book Store\"/>");
out.println("<var name=\"added_cost\" />");
out.println("<var name=\"added_items\" expr=\""+num_of_items+"\" />");
out.println("<form id=\"calculate_price\">");
out.println("  <subdialog name=\"discount\"\"");
out.println("    src=\"./CalculateDiscount.vxml#calculate_discount\"");
out.println("    <param name=\"quantity\"");
out.println("        expr=\"application.current_item=='book' ?\"");
out.println("            application.total_books : ");
out.println("            application.total_shirts\" />\"");
out.println("    <param name=\"unit_price\"");
out.println("        expr=\"application.current_item=='book' ?\"");
out.println("            application.unit_book_price : ");
out.println("            application.unit_tshirt_price\" />\"");
out.println("    <param name=\"isCoupon\"");
out.println("        expr=\"application.current_item=='book' ? \"");
out.println("            application.has_coupon : false\" />\"");
out.println("  <filled>");
out.println("    <if cond=\"application.current_item == 'book'\">");
out.println("      <assign name=\"application.total_book_cost\"\"");
out.println("          expr=\"Number(discount.price)\"/>\"");
out.println("    </if>\"");
out.println("    <assign name=\"application.total_tshirt_cost\"\"");
out.println("        expr=\"Number(discount.price)\"/>\"");
out.println("  </if>\"");
out.println("  <!-- Keep track of the cost of the books added -->\"");
out.println("  <var name=\"old_cost\"\"");
out.println("      expr=\"application.total_order_cost\"/>\"");
out.println("  <assign name=\"application.total_order_cost\"\"");
out.println("      expr=\"application.total_book_cost\"");
out.println("          + application.total_tshirt_cost\"/>\"");
out.println("  <!-- Cost of the books added=new total-old total-->\"");
out.println("  <assign name=\"added_cost\"\"");
out.println("      expr=\"application.total_order_cost - \"");
out.println("          old_cost\"/>\"");
out.println("  <goto next=\"#output_total\" />\"");

```

```

out.println("      </filled>");
out.println("    </subdialog>");
out.println("</form>");
out.println("<form id=\"output_total\">");
out.println("<property name=\"fetchtimeout\" value=\"2s\"/>");
out.println("  <block>");
out.println("    <audio src=\"youhave.wav\"> You have </audio>");
out.println("    <audio src=\"added.wav\"> added </audio>");
out.println("    <value expr=\"String(Math.round(added_items))\"/>");
out.println("    <!-- output book, books, tshirts, or tshirts -->");
out.println("    <prompt cond=\"added_items == 1 &&\"");
out.println("      application.current_item == 'book'\">");
out.println("      <audio src=\"book.wav\">book</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items != 1 &&\"");
out.println("      application.current_item == 'book'\">");
out.println("      <audio src=\"books.wav\">books</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items == 1 && \"");
out.println("      application.current_item== 'shirt'\">");
out.println("      <audio src=\"tshirt.wav\">t-shirt</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items != 1 && \"");
out.println("      application.current_item== 'shirt'\">");
out.println("      <audio src=\"tshirts.wav\">t-shirts</audio>");
out.println("    </prompt>");
out.println("    <audio src=\"to.wav\">to </audio>");
out.println("    <audio src=\"yourcart.wav\">");
out.println("      your shopping cart. ");
out.println("    </audio>");
out.println("    <audio src=\"thecost.wav\">The cost of the </audio>");
out.println("    <value expr=\"String(Math.round(added_items))\"/>");
out.println("    <audio src=\"added.wav\">added</audio>");
out.println("    <!-- output book, books, tshirts, or tshirts -->");
out.println("    <prompt cond=\"added_items == 1 &&\"");
out.println("      application.current_item == 'book'\">");
out.println("      <audio src=\"book.wav\">book</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items != 1 && \"");
out.println("      application.current_item == 'book'\">");
out.println("      <audio src=\"books.wav\">books</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items == 1 && \"");
out.println("      application.current_item== 'shirt'\">");
out.println("      <audio src=\"tshirt.wav\">t-shirt</audio>");
out.println("    </prompt>");
out.println("    <prompt cond=\"added_items != 1 && \"");
out.println("      application.current_item== 'shirt'\">");
out.println("      <audio src=\"tshirts.wav\">t-shirts</audio>");
out.println("    </prompt>");
out.println("    <audio src=\"is.wav\">is </audio>");
out.println("    <value expr=\"added_cost\" class=\"currency\" />.");

```

```

        out.println("      <goto next=\"ListItems.vxml\" fetchtimeout=\"4s\" />");
        out.println("    </block>");
        out.println("  </form>");
        out.println("</vxml>");
        out.close();
    }
}

<?xml version="1.0"?>
<!--ListItems.vxml-->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright"
        content="Sams Publishing 2001"/>
  <meta name="description"
        content="List the content of the shopping cart,
        to access call (xxx)-xxx-xxxx"/>
  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <meta name="maintainer" content="johnoreilly@lucent.com"/>
  <meta name="application" content="Book Store"/>
  <form id="output_total">
    <property name="fetchtimeout" value="2s"/>
    <block>
      <audio src="cartcontains.wav">
        Your shopping cart now contains
      </audio>
      <value expr="String(Math.round(application.total_books))" />
      <prompt cond="application.total_books == 1">
        <audio src="book.wav">book</audio>
      </prompt>
      <prompt cond="application.total_books != 1">
        <audio src="books.wav">books</audio>
      </prompt>
      <audio src="and.wav"> and </audio>
      <value expr="String(Math.round(application.total_shirts))" />
      <prompt cond="application.total_shirts == 1">
        <audio src="tshirt.wav">t-shirt</audio>
      </prompt>
      <prompt cond="application.total_shirts != 1">
        <audio src="tshirts.wav">t-shirts</audio>
      </prompt>

      <audio src="totalcost.wav">
        The total cost of the items in your shopping cart is
      </audio>
      <value expr="application.total_order_cost" class="currency" />
      <goto next="WelcomeMenuProp.vxml#what_next" fetchtimeout="4s" />
    </block>
  </form>
</vxml>

```

```

        </form>
    </vxml>

    <?xml version="1.0"?>
    <!--RemoveItem.vxml-->
    <vxml version="1.0" application="BookStoreRoot.vxml" >
        <meta name="author" content="John O'Reilly"/>
        <meta name="copyright"
              content="Sams Publishing 2001"/>
        <meta name="description"
              content="Select Item to remove, to access call (xxx)-xxx-xxxx"/>
        <meta name="keywords" content="VoiceXML Sams Books"/>
        <meta name="maintainer" content="johnnny@sams.net"/>
        <meta name="robots" content="index,follow"/>
        <meta http-equiv="Date"
              content="Thu, 7 Aug 2000 20:20:20 EST"/>
        <meta name="maintainer" content="johnoreilly@lucent.com"/>
        <meta name="application" content="Book Store"/>
    <form id="pick_item">
        <field name="item">
            <prompt>
                <enumerate>
                    To remove <value expr="_prompt"/>
                    from your shopping cart, say <value expr="_prompt"/>
                    or press <value expr="_dtmf"/>.
                </enumerate>
            </prompt>
            <option dtmf="1" value="book"> books </option>
            <option dtmf="2" value="shirt"> tee shirts </option>
            <filled>
                <assign name="application.current_item" expr="item" />
                <goto next="RemoveStuff.vxml" />
            </filled>
            <help cond="global_help">
                You are currently selecting an item to remove from your
                shopping cart. Say the name of, or press the phone key
                of the item you would like to remove.

                <assign name="global_help" expr="false" />
                <throw event="help" />
            </help>
        </field>
    </form>
</vxml>

    <?xml version="1.0"?>
    <!--RemoveStuff.vxml-->
    <vxml version="1.0" application="BookStoreRoot.vxml">
        <meta name="author" content="John O'Reilly"/>
        <meta name="copyright" content="Sams Publishing 2001"/>
        <meta name="description"
              content="Remove j items to a shopping cart,
              to access call (xxx)-xxx-xxxx"/>

```

```
<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>

<form id="collect_items">
    <property name="bargein" value="true"/>
    <property name="timeout" value="4s"/>
    <catch event="error.invalid_quantity">
        I'm sorry, <value expr="num_of_items" /> is not a valid entry.
        Order quantities can not be fractions or negative.
        Let's try again.
        <clear namelist="num_of_items confirm" />
        <goto nextitem="num_of_items" />
    </catch>

    <catch event="error.too_many">
        I'm sorry, the <value expr="num_of_items" />
        <value expr="application.current_item +
                    (num_of_items != 1 ? 's' : '')"/>
        you have asked to remove from your shopping cart is more than the
        <value expr="(application.current_item=='book' ?
                    application.total_books :
                    application.total_shirts)" />
        <value expr="application.current_item +
                    (num_of_items != 1 ? 's' : '')"/>
        already in your shopping cart.
        Let's try again.
        <clear namelist="num_of_items confirm" />
        <goto nextitem="num_of_items" />
    </catch>

    <field name="num_of_items" type="number">
        <prompt>
            How many <value expr="application.current_item + 's'" />
            would you like to remove from your shopping cart?
        </prompt>
        <catch event="help" cond="global_help">
            Say or enter using the number keys on your phone the quantity
            of <value expr="application.current_item + 's'" /> you wish to
            remove from your shopping cart.

            <assign name="global_help" expr="false" />
            <throw event="help" />
        </catch>
    </field>

    <field name="confirm" type="boolean?y=1;n=0">
        <prompt>
            You have asked to remove
            <value expr="num_of_items" /> <!-- number of items -->

```

```
<!-- singular or plural of the item being added -->
<value expr="application.current_item +
    (num_of_items != 1 ? 's' : '')"/>
from your shopping cart. Is this correct?
</prompt>
<catch event="help" cond="global_help">
    If the quantity you entered is correct, say yes or press the
    one key on your phone.
    If you would like to reenter the quantity of
    <value expr="application.current_item +
        (num_of_items != 1 ? 's' : '')"/> to remove,
    say no or press the zero key on your phone.
    <assign name="global_help" expr="false" />
    <throw event="help" />
</catch>
</field>
<filled namelist="num_of_items confirm">
    <!-- Only whole positive quantities will be be excepted -->
    <!-- Make sure the user entered an integer -->
    <var name="is_fraction"
        expr="Math.ceil(num_of_items) != Math.floor(num_of_items)" />
    <var name="is_negative" expr="num_of_items < 0" />

    <if cond="confirm == false">
        <clear namelist="num_of_items confirm" />
    <elseif cond="num_of_items == 0"/>
        <prompt>
            You have chosen not to remove any
            <value expr="application.current_item + 's'" />
            from your shopping cart.
        </prompt>
        <goto next="WelcomeMenuProp.vxml#what_next" />
    <!-- Make sure the user entered an integer -->
    <elseif cond="is_fraction || is_negative" />
        <throw event="error.invalid_quantity" />
    <!-- Make sure the user does not try to remove more
        than he or she has -->
    <elseif cond="num_of_items >
        (application.current_item=='book' ?
            application.total_books :
            application.total_shirts)" />
        <throw event="error.too_many" />
    <else/>
        <if cond="application.current_item=='book'" >
            <assign name="application.total_books"
                expr="application.total_books -
                    Number(num_of_items)" />
        <else/>
            <assign name="application.total_shirts"
                expr="application.total_shirts -
                    Number(num_of_items)" />
        </if>
```

```

        <submit next=".//servlets/CalculateCostRemove"
               namelist="num_of_items" fetchtimeout="3s" />
    </if>
</filled>
</form>
</vxml>

import javax.servlet.*;
import javax.servlet.http.*;
import java.text.*;
import java.io.*;
import java.net.*;

public class CalculateCostRemove extends HttpServlet
{
    public void doGet( HttpServletRequest req, HttpServletResponse res )
                     throws ServletException, IOException
    {
        res.setContentType( "text/xml" );
        ServletOutputStream out = res.getOutputStream();

        String num_of_items = req.getParameter("num_of_items");

        out.println("<?xml version='1.0'?'>");
        out.println("<!--CalculateCostRemove.vxml-->");
        out.println("<vxml version='1.0'" );
        out.println("      application='BookStoreRoot.vxml'");
        out.println("<meta name='author' content='John O'Reilly'> ");
        out.println("<meta name='copyright'" );
        out.println("      content='Sams Publishing 2001'> ");
        out.println("<meta name='description'" );
        out.println("      content='Calculate the cost of an order, '");
        out.println("              to access call (xxx)-xxx-xxxx'> ");
        out.println("<meta name='keywords' content='VoiceXML Sam Books'> ");
        out.println("<meta name='maintainer' content='johnnyo@sam.net'> ");
        out.println("<meta name='robots' content='index,follow'> ");
        out.println("<meta http-equiv='Date'" );
        out.println("      content='Thu, 7 Aug 2000 20:20:20 EST'> ");
        out.println("<meta name='maintainer' " );
        out.println("      content='johnoreilly@lucent.com'> ");
        out.println("<meta name='application' content='Book Store'> ");
        out.println("<property name='caching' value='safe' /> ");
        out.println("<var name='removed_items' expr='"+num_of_items+"'/> ");
        out.println("<form id='calculate_price'" );
        out.println("   <subdialog name='discount'" );
        out.println("     src='./CalculateDiscount.vxml#calculate_discount'> ");
        out.println("       <param name='quantity' " );
        out.println("             expr='application.current_item=='book' ? ' ");
        out.println("                           application.total_books : ' ");
        out.println("                           application.total_shirts' /> ");
        out.println("       <param name='unit_price' " );
        out.println("             expr='application.current_item=='book' ? ' ");
        out.println("                           application.unit_book_price : ' ");
    }
}

```

```

out.println("           application.unit_tshirt_price\" />\"");
out.println(" <param name=\"isCoupon\" \"");
out.println("   expr=\"application.current_item=='book' ? \"");
out.println("           application.has_coupon : false\" />\"");
out.println(" </filled>\"");
out.println(" <if cond=\"application.current_item == 'book'\">>\"");
out.println("   <assign name=\"application.total_book_cost\"\"");
out.println("     expr=\"Number(discount.price)\" />\"");
out.println(" </else/>\"");
out.println("   <assign name=\"application.total_tshirt_cost\"\"");
out.println("     expr=\"Number(discount.price)\" />\"");
out.println(" </if>\"");
out.println(" <assign name=\"application.total_order_cost\"\"");
out.println("   expr=\"application.total_book_cost +\"");
out.println("           application.total_tshirt_cost\"/>\"");
out.println(" <goto next=\"#output_total\"/>\"");
out.println(" </filled>\"");
out.println(" </subdialog>\"");
out.println("</form>");

out.println("<form id=\"output_total\">");
out.println(" <property name=\"fetchtimeout\" value=\"2s\"/>\"");
out.println(" <block>\"");
out.println("   <audio src=\"youhave.wav\">You have</audio>\"");
out.println("   <audio src=\"removed.wav\">removed</audio>\"");
out.println("   <value expr=\"String(Math.round(removed_items))\"/>\"");
out.println("   <!-- output book, books, tshirts, or tshirts -->\"");
out.println("   <prompt cond=\"removed_items == 1 && \"\"");
out.println("     application.current_item == 'book'\">>\"");
out.println("     <audio src=\"book.wav\">book</audio>\"");
out.println("   </prompt>\"");
out.println("   <prompt cond=\"removed_items != 1 && \"\"");
out.println("     application.current_item == 'book'\">>\"");
out.println("     <audio src=\"books.wav\">books</audio>\"");
out.println("   </prompt>\"");
out.println("   <prompt cond=\"removed_items == 1 && \"\"");
out.println("     application.current_item == 'shirt'\">>\"");
out.println("     <audio src=\"tshirt.wav\">t-shirt</audio>\"");
out.println("   </prompt>\"");
out.println("   <prompt cond=\"removed_items != 1 && \"\"");
out.println("     application.current_item == 'shirt'\">>\"");
out.println("     <audio src=\"tshirts.wav\">t-shirts</audio>\"");
out.println("   </prompt>\"");
out.println("   <audio src=\"from.wav\">from</audio>\"");
out.println("   <audio src=\"yourcart.wav\"> \"");
out.println("     your shopping cart. \"");
out.println("   </audio>\"");
out.println("   <goto next=\"ListItems.vxml\" fetchtimeout=\"4s\" />\"");
out.println(" </block>\"");
out.println(" </form>\"");
out.println("</vxml>\"");
out.close();

```

```

}

}

<?xml version="1.0"?>
<!--GetComments.vxml-->
<!--This document demonstrates the record command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
<property name="caching" value="safe" />
<meta name="author" content="John O'Reilly"/>
<meta name="copyright" content="Sams Publishing 2001"/>
<meta name="description"
      content="This page collects comments for Sams online bookstore,
      to access call (xxx)-xxx-xxxx"/>
<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
<form id="record_comments">
  <record name="comments" modal="true" beep="true" maxtime="60s"
         finalsilence="2s" dtmfterm="true">
    <prompt>
      Please enter your comments at the beep.
    </prompt>
    <filled>
      <prompt>
        The comments <value expr="comments" mode="recorded"/> have been saved.
      </prompt>
      <submit next="http://www.myisp.net/beez/servlet/RecordComments"
             method="post" namelist="comments" />
    </filled>
  </record>
</form>
</vxml>
import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import stec.servlet.*;

public class RecordComments extends HttpServlet
{
  public void doPost( HttpServletRequest req,
                     HttpServletResponse res )
    throws ServletException, IOException
  {
    ServletOutputStream out = res.getOutputStream();
    String output = "Let's return to the main menu.";
    res.setContentType( "text/xml" );

    try
    {
      ServletContext context = getServletContext();

```

```
String basedir = context.getRealPath( "./audio/" );
File f = new File( basedir );
f.mkdirs();

Hashtable ht = MultiPartForm.parse( req,
    f.getCanonicalPath(), 1024*1024 );
}

catch( Exception e )
{
    output = e.toString();
}

out.println("<?xml version=\"1.0\"?> " );
out.println("<!--UploadComments.vxml-->");
out.println("<vxm version=\"1.0\""
    application=\"BookStoreRoot.vxml\" >");
out.println("<meta name=\"author\" content=\"Rick Beasley\"/>");
out.println("<meta name=\"copyright\"\"");
out.println("      content=\"Sams Publishing 2001\"/>");
out.println("<meta name=\"description\"\"");
out.println("      content=\"Upload a recorded audio file, \"");
out.println("          to access call (xxx)-xxx-xxxx\"/>\"");
out.println("<meta name=\"keywords\""
    content=\"VoiceXML Sams Books\"/>\"");
out.println("<meta name=\"maintainer\""
    content=\"beez@sams.net\"/>\"");
out.println("<meta name=\"robots\" content=\"index,follow\"/>\"");
out.println("<meta http-equiv=\"Date\"\"");
out.println("  content=\"Thu, 7 Aug 2000 20:20:20 EST\"/>\"");
out.println("<meta name=\"maintainer\" \"\"");
out.println("      content=\"johnoreilly@lucent.com\"/>\"");
out.println("<meta name=\"application\""
    content=\"Book Store\"/>\"");
out.println("  <form> ");
out.println("    <block>\"");
out.println("      +output");
out.println("      <goto"
        next=\"http://localhost/WelcomeMenuProp.vxml\"/>\"");
out.println("    </block> ");
out.println("  </form> ");
out.println("</vxm>");

}

<?xml version="1.0"?>
<!--CheckOutScript.vxml-->
<!--This document demonstrates the script element -->
<vxm version="1.0">
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright" content="Sams Publishing 2001"/>
    <meta name="description"
        content="Find out how long it will take to ship the items,
```

```
        to access call (xxx)-xxx-xxxx" />
<meta name="keywords" content="VoiceXML Sams Books" />
<meta name="maintainer" content="johnnyo@sams.net" />
<meta name="robots" content="index,follow" />
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST" />

<form id="get_location">
    <grammar src="statecountry.g" type="application/x-jsgf" />

    <initial name="start">
        <prompt>
            Please say the state, if in the United States, or the
            country you would like to ship your order to.
        </prompt>
        <help cond="global_help">
            You have chosen to check out.
            You are currently being asked for the shipping address for
            your order.
            <assign name="global_help" expr="false" />
            <throw event="help" />
        </help>
    </initial>
    <field name="state">
        <prompt>Which state would you like to send the order to? </prompt>
        <filled>
            <assign name="country" expr="'United States'" />
        </filled>
    </field>

    <field name="country">
        <filled>
            <if cond="country != 'United States'">
                <assign name="state" expr="null"/>
            </if>
        </filled>
    </field>

    <field name="confirm" type="boolean?y=1;n=0">
        <prompt> You have requested to send your order to </prompt>
        <prompt cond="state != null">
            <value expr="state" />,
        </prompt>
        <prompt>
            <value expr="country" />
            Is this correct?
        </prompt>
        <filled>

            <var name="day" />

            <var name="spoken_month" />
            <if cond="confirm == true">
```

```
<script>
    date = new Date();
    date.setDate(date.getDate() + 14);
    month = date.getMonth();
    day = date.getDate();
    switch (month)
    {
        case 0:
            spoken_month = "January";
            break;
        case 1:
            spoken_month = "February";
            break;
        case 2:
            spoken_month = "March";
            break;
        case 3:
            spoken_month = "April";
            break;
        case 4:
            spoken_month = "May";
            break;
        case 5:
            spoken_month = "June";
            break;
        case 6:
            spoken_month = "July";
            break;
        case 7:
            spoken_month = "August";
            break;
        case 8:
            spoken_month = "September";
            break;
        case 9:
            spoken_month = "October";
            break;
        case 10:
            spoken_month = "November";
            break;
        case 11:
            spoken_month = "December";
            break;
    }
</script>

<prompt>
    Your order has been processed. It will arrive by
    <value expr="spoken_month" />
    <value expr="String(Math.round(day))" />.
    Thank you for ordering from Sams Online Bookstore.
</prompt>
```

```
<else/>
    <clear namelist="start country state confirm" />
</if>
</filled>
</field>
</form>
</vxml>

#JSGF V1.0;
grammar stateandcountry;
public <statecountry> =
    <state> {this.state=$} [<country> {this.country=$}] |
    <country> {this.country=$};

<state> = Alabama | Alaska | Arizona | Arkansas | California | Colorado |
    Connecticut | Delaware | Florida | Georgia | Hawaii | Idaho |
    Illinois | Indiana | Iowa | Kansas | Kentucky | Louisiana | Maine |
    Maryland | Massachusetts | Michigan | Minnesota | Mississippi |
    Missouri | Montana | Nebraska | Nevada | New Hampshire | New Jersey |
    New Mexico | New York | North Carolina | North Dakota | Ohio |
    Oklahoma | Oregon | Pennsylvania | Rhode Island | South Carolina |
    South Dakota | Tennessee | Texas | Utah | Vermont | Virginia |
    Washington | West Virginia | Wisconsin | Wyoming;
<country> = United States | Canada | Mexico;
```

附录C VoiceXML 2.0的有关情况

VoiceXML 1.0是VoiceXML论坛建立的一个技术标准。VoiceXML的一个新版本，即人们通常所说的VoiceXML 2.0，目前正在由万维网联盟（W3C）的语音浏览器工作组进行开发。VoiceXML 2.0目前处于工作草案的开发阶段，还没有向公众推出。

VoiceXML 2.0不包括与语法和语音标记相关的VoiceXML 1.0的函数。所谓的VoiceXML 2.0将包括：

- VoiceXML——语音浏览器组件。
- 语音合成标记语言（SSML）——提供合成语音标记的函数。
- 语音识别语法技术规范（SRGS）——用于定义代表语音识别语法的文法。

我们预计VoiceXML 2.0将会在下列方面对VoiceXML 1.0进行较小的修改：

- 增加——新的元素、属性、特征、出错条件和事件。我们预计这方面的大部分修改将采用给VoiceXML 1.0已经设定的元素增加新属性的形式。
- 修改——对VoiceXML 1.0中的某些东西进行删除和更改。
- 澄清——VoiceXML 2.0的技术规范将比VoiceXML 1.0更加清楚。

当这个新的技术规范向公众推出时，我们将在网址 www.sampublishing.com 上放置一个完整的在线附录。同时，你可以通过网址 <http://www.w3.org/voice/> 来关注W3C的语音浏览器工作组从事的与VoiceXML 2.0有关的活动。