

智能客户端在 Microsoft Word System 系统中的集成实现

丰晋军 尹战文 (江南计算技术研究所 214083)

摘要:在基于企业环境传统的客户端/服务器两层结构的应用程序模型当中,c/s(客户端/服务器模型),b/s(浏览器/服务器模型)两种方式面临着世代替换,而基于.net平台的智能客户端应用程序具备了丰富型客户端模型的强大功能和弹性,还有以基于浏览器模型的易于不部署和稳定性等功能,而本文讨论了 Microsoft word System 与 xml 的关系以及如何在 Microsoft word System 中智能客户端的实现形式,为企业以 Word 为客户端程序模型实现智能客户端提供了一套优良的解决方案。

关键词:XML 架构 客户端模型 XML 扩展包 智能文档 智能标签

1 引言

智能客户端是一种易于部署和管理的客户端应用程序,它通过利用本地资源和智能化连接到分布式数据源,提供了适应性和交互性的体验。智能客户端有三种表现形式。分别是 Windows Application,智能设备应用和 Smart document 的应用。由于智能客户端是作为 Microsoft .net 平台的一个特性而被提出的,.NET 从系统和语言级别对它提供了支持,实际上,智能客户端是一个可扩展的能继承不同应用的桌面应用程序,它具备充分利用本地资源,丰富的用户交互,支持联机、离线操作,智能部署和更新,安全性,设备适应性等特性。

2 Word 系统与 XML 语言

随着 Microsoft Office System 2003 的出现,在 Office System 中几乎所有的重要组件都加强了对 xml 的支持,Word 2003 支持创建、编辑、转换和保存 XML 格式文件。XML 已经成为 Office System 中非常重要的一项内容,这既加强了 Office 自身各组件的信息交换能力,与此同时也对于外来信息的搜集、处理以及发布做出了良好的准备。

Word 2003 针对 XML 进行了完成的设计,它支持称为 Word 标记语言 (WordML) 的原生 XML 词汇,WordML 的功能非常强大和灵活,Word 2003 还可以将自定义的 XML 架构定义和 XSLT 转换附加到 Word 文

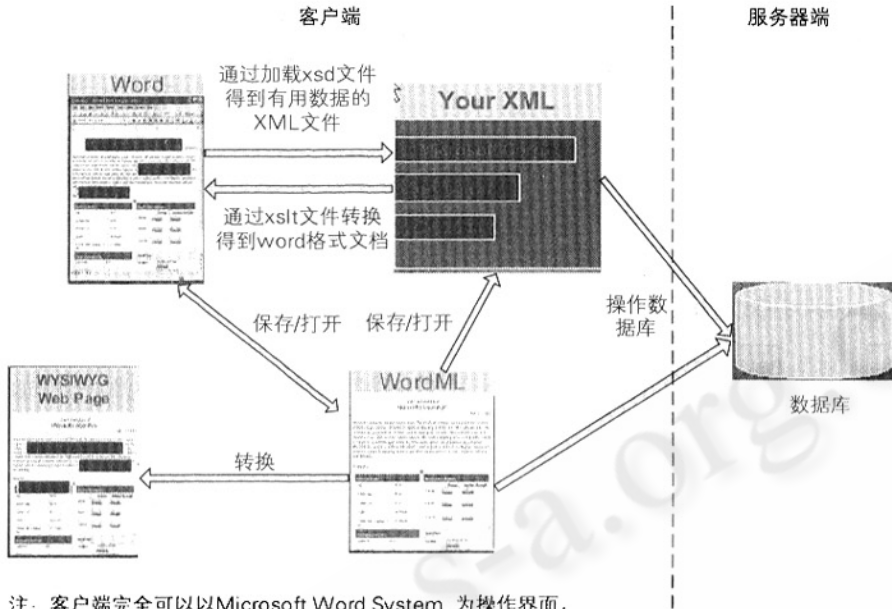
档。开发人员可以利用附加架构中的元素来标记内容,这样就可以插入用于简化处理过程的有意义的业务特定标记。当您在 Word 中保存文档时,它可以根据附加的架构来验证文档,并可以在处理过程中应用自定义的 XSLT 转换。示意图如图 1 所示。

在新的 Word 系统中,当添加了 xml 支持以后,对 WordML 的原生支持,对 XML 架构,实时验证以及各种 XSLT 技术的内置支持,使得利用自定义的 XML 架构和输出格式能够轻松的创建各种 Word 解决方案。

3 智能文档在 Word 系统中的实现

3.1 智能文档概述

随着 Windows 中出现越来越多的功能以及企业用户和个人需要更有力的软件来帮助他们管理身边的任务,智能客户端技术开始变得日趋重要。智能文档是开发用于将业务逻辑和自定义操作绑定到 Office Word 2003 文档内容的解决方案。Microsoft Office 2003 文档不仅仅是用户输入的信息的储存库,还可以通过编程使它们变为“智能的”文档。智能文档可以自动输入数据,这样信息工作者就可以更容易地填写表单或使用模板。智能文档也可以自动访问外部数据并将其放入文档中的相应位置,或者提供上下文帮助以指导信息工作者准备复杂的文档。智能文档中的信息甚至可以在完全不同的系统和应用程序之间共享。Microsoft Office Word



注：客户端完全可以以Microsoft Word System 为操作界面，然后在Action Pane中做程序设计处理

图 1 Xml 和 word 关系示意图

2003 中提供的智能文档包含编程逻辑,该编程逻辑定义文档的使用方式,并控制文档中数据的操作方式。智能文档从对 XML 代码的内置支持获得其功能,这些代码是智能文档操作的基本平台。XML 元素包含在文档中,它们定义和启动相关联的操作。可以用多种编程语言编写这些控制后续操作的代码,这使得开发人员可以为任何业务进程设计可行的解决方案。

3.2 智能文档的特征与应用

究竟什么是智能文档呢?所谓智能文档,就是一种动态容器,以支持 xml 语言的办公软件(word2003, excel2003, Adobe 等)为操作平台,借助嵌入式可执行后台代码参与到程序流程之中。智能文档主要使用 XML 语言,而 XML 则可以通过假定不可知的方式显示数据类型,并执行数据的采集,集成,格式转换等等。智能文档是业务流程 + 信息、存档以及交互性的知识库,是动态的,可以直接跟其它的内容、数据、应用甚至文档进行交互。它将传统的电子文档与信息系统融合起来。智能文档的本质是使用真正的“文档”构造常见的应用系统,使得软件模型和用户的现实业务模型得到精确的匹配。正是由于这些特性,使得智能文档在电子公文、办公自动化、企业申报审批、政府门户服务、企业信息管理集成、文档共享管理等应用中有良好的技术支持。

3.3 智能文档的用法

智能文档可为文档的指定部分(使用 XML 标记分隔)分配操作或控件。在这些部分中,通过“文档操作”任务窗格,智能文档可以向信息工作者提供图形、帮助文本、文档片段或基于控件(例如文本框、单选按钮、命令按钮,甚至自定义的 ActiveX 控件)的操作形式的信息。

3.4 智能文档的结构

使用智能文档,开发人员可以将指定操作分配给以前已存在的或新创建的文档内的 XML 元素。映射文档内的 XML 与基本的 XML 架构相对应。文档准备完毕

后,开发人员可以使用智能文档 API(作为 Office 2003 Solution Developers Kit 的一部分提供)来分配将驱动解决方案的操作。使用智能文档时,开发人员有很大的灵活性。他们可以直接操作文档或与服务器端进程交互,例如检索数据或将数据路由到其他位置以供后端系统使用。在如何开发方面开发人员也有很大的灵活性:他们可用使用 Visual Basic 6.0、Visual Basic .NET、C# 或 C++。编译完智能文档 DLL 后,开发人员必须创建一个清单文件以描述下列项目的位置:

- 包含驱动解决方案的自动化代码的 DLL 或者 Microsoft Word System 智能标签列表文档
- 与文档本身中的映射元素相对应的 XML 架构
- 帮助对信息工作者屏蔽复杂结构的 XML 清单
- 其他文件(如图片文件,网页文件等)

3.5 智能文档的开发方法

开发智能文档有两种方法:

(1) 自己创建链接到文档的 DLL 链接库程序,这种智能文档解决方案提供了最大的自由度,但是工作量相对来说要大一些;

(2) 可以使用 XML 编辑器(如 Microsoft 记事本或 Word 2003)创建 XML 文件类型的智能文档解决

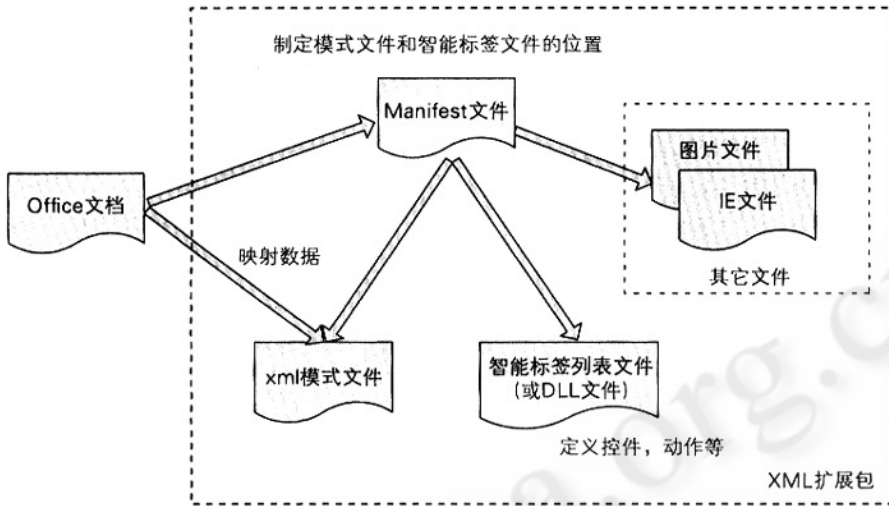


图 2 智能文档的结构图

方案,由于 Word 2003 和 Excel 2003 内置了对标准 Action 的支持,所以通过简单的编辑 XML 描述文件,便可以创建出比较简单的智能文档解决方案,但是这种方法创建的智能文档,其操作仅仅局限于静态文本、超链接、文档片段和图像,因此,这种通用的智能化模型对于单个具体的商业应用领域实用性不大。无论采用这两种方法中的哪一种,智能文档开发流程都很简单。

3.6 智能文档开发步骤

智能文档开发的步骤如下:

3.6.1 设计 Schema 架构文件,制作 Word 模版文件,并为其添加 XML 架构支持

智能文档功能需要建立在 XML 格式的架构文件之上。因此,对于自定义的智能文档解决方案,需要自己定义架构文件(schema)。架构文件的定义要根据具体的需求分析结果来定义。编写 Word 格式的文件,为其添加自定义的 XML 架构,并把相应的 XML 元素映射到文档中。

3.6.2 开发智能文档链接库程序或智能标签列表文件,对 XML 架构定义的元素定义相关智能操作

基于 MOSTL 文件(智能标签列表)的智能文档开发比较简单,但其操作仅仅局限于静态文本、超链接、文档片段和图像等,有很大的局限性,在此义基于 DLL 的智能文档为例,来阐述智能文档的代码开发过程。开发智能文档链接库文件可以采用多种通用的高级编

程语言,比如 C++、Visual Basic 语言以及 .Net 平台下的各种开发语言,C#、VB、Net、J#等。就开发的效率和安全性来说推荐使用 .Net 平台下的 C#或者 VB、Net 语言进行智能文档的开发。简单说就是编写代码实现 "ISmartDocument" 接口。下面以在 ActionPane 中加入 Button 控件为例,开发过程大概如下:

- 通过 "SmartDocXmlTypeCount" "SmartDocXmlTypeName" "Smart-

DocXmlTypeCaption" 得到 xml 架构文件中的 element 元素的 Count, Name, 和 Caption。

- "ControlCount" 属性指定每个 namespace#element 名称可使用多少个控件。因为我们添加了 BUTTON 控件,所以要在原来的返回值上再加一。"ControlID" 为 BUTTON 控件分配一个唯一的 ID 号。

- "ControlNameFromID" 属性为每个控件 BUTTON 分配一个唯一的名称,以便以后需要时可以从 Word 或 Excel 应用程序对象模型中引用它。例: public string get_ControlNameFromID (int ControlID) { return cNAMESPACE + ControlID. ToString (); }

- "ControlCaptionFromID" "ControlTypeFromID" 可以得到 BUTTON 的标题名和类型,类型为 C_TYPE. C_TYPE. BUTTON。

- 调用这些控件的特定初始化方法和特定操作方法,例

```
public void InvokeControl (int ControlID, string ApplicationName, object Target, string Text, string Xml, int LocaleID)
```

```
{ switch ( ControlID)
{
```

```
case 101://通过 ControlID 判断是否是 BUTTON 控件
System. Windows. Forms. MessageBox. Show
(" This is an example of a button. " );
```

```
break;
```

```
default:
break;
}}
```

3.6.3 编写 Manifest 文件,用于为定义好的智能文档模版加载智能文档解决方案

完成了智能文档链接库文件之后,接下来要编写一个 XML 扩展包清单文件(Manifest.xml),可以使用用户界面引用 XML 扩展包清单文件并将该解决方案附加到文档或工作簿。该 XML 清单描述文件用于描述智能文档解决方案的架构和链接库文件存放的位置、版本等信息,利用这个清单文件宿主程序(Word)将可以自动加载智能文档扩展包程序,这样就可以直接使用文档的智能文档功能了。文档头一般定义以 SD 为前缀的命名空间,然后定义了 Manifest 文电的版本更新频率和 URI,而文档内容呢,由一个或多个 solution 元素组成,在 solution 元素中,有 solutionID, version, type, file 元素组成,而有根据 type 的值不同, solution 元素有着不同的必需元素,而 file 元素中又有 filepath, type, version 元素组成,下例是一个 manifest 文件片断:

```
<? xml version="1.0" encoding="utf-8" ? >
<SD:manifest xmlns:SD="http://schemas.microsoft.com/office/xmlexpansionpacks/2003" >
  <SD:version >1.2 </SD:version >
  <SD:updateFrequency >20160 </SD:updateFrequency >
  <SD:url >acerfjicodebook </SD:url >
  <SD:solution >
    <SD:solutionID > {12341234-1234-1234-1234-123456789012} </SD:solutionID >
    <SD:type > smartDocument </SD:type >
    <SD:alias lcid="*" > Mr.Feng'scodebook
  </SD:alias >
    <SD:file >
  <SD:type > solutionActionHandler </SD:type >
    <SD:version >1.2 </SD:version >
  <SD:filePath > D:\phonecode\bin\Debug\
```

```
phonecode.dll </SD:filePath >
  <SD:CLSNAME > acerfjicodebook.CodeNum </SD:CLSNAME >
  <SD:managed/ >
  </SD:file >
  ...
</SD:solution >
<SD:solution >
  ...
</SD:manifest >
```

4 总结

智能文档利用了 XML 带给 Office 2003 文档的易理解性,以便实现对企业数据存储的无缝、往返访问。智能文档允许单位以预定的方式使用 XML。构建在 Microsoft Office 系统程序之上的智能解决方案可以动态访问 Web 服务,进行数据的分析、共享和处理。这样得到的是灵活、可自定义且总是具有最新信息的应用程序,它们在联机 and 脱机情况下都可用,而且在脱机使用时仍可保留所有用户功能。通过使用熟悉的 Office 应用程序和 Microsoft Office 系统提供的领先技术来创建智能客户端,企业还可以利用在信息工作者培训和技术方面的现有投资。这使得企业可以使用 Microsoft Office 系统创建智能客户端解决方案,以便确保及时更新企业信息。

参考文献

- 1 Peter Kelly. Microsoft Office 系统中的智能客户端应用程序概述, 2001 Microsoft Office System
- 2 Stewart Fraser, Steven Livingstone. C#XML 入门经典[M], 北京 清华大学出版社, 2003。
- 3 张晓东, Office 2003 中的文档智能化操作, Msdn.com。
- 4 Microsoft Office 2003, 摘要技术白皮书, Microsoft Office System, 2003 年 7 月。
- 5 Evan Lenz, Mary McRae, Simon St. Laurent. Office 2003 XML[C] O'reilly, June 2003.