

ActiveX 在区域森林资源预警 技术系统中的应用^{*}

吴延熊

周国模 郭仁鉴

(云南省林业科学院重点实验室, 昆明 650204) (浙江林学院资源与环境系)

摘要 论述了区域森林资源预警技术系统开发的核心技术之一的 ActiveX 技术。从 ActiveX 与软件危机关系的角度, 讨论了 ActiveX 组件技术的重要性, 并对 ActiveX 组件技术的基本原理及其开发工具的支持情况作了简单介绍。举例说明了预警技术系统开发中 ActiveX 组件的创建和应用, 并且比较了与 ActiveX 相互竞争的 2 种技术: CORBA 和 Java Bean。图 2 参 3

关键词 区域; 森林资源; 预警技术系统; Delphi 4. 0; ActiveX 组件; 应用
中图分类号 S757. 1; Q-0

区域森林资源预警技术系统是一类比较复杂的信息系统, 而且目前还处在试制和探索阶段, 因此所有的开发工作不需要也不可能从底层做起, 而应选择一个较高的起点, 充分利用现今先进的计算机技术, 特别是面向对象技术。

我们研制的县域森林资源预警技术系统 RFREWS 1. 0 充分利用了面向对象技术核心之一的 ActiveX 技术。

1 ActiveX 与软件危机

1964 年 ENIAC 的诞生标志着人类拥有了以真空管继电器为基础的第 1 代计算机, 此后计算机技术突飞猛进, 更新换代的周期日益缩短。CPU 处理速度、硬盘容量、内存大小以及整体可靠性都有了惊人的提高, 而成本价格却不断下降。特别是低价高性能的微型计算机和方便快捷的信息高速公路的发展, 使得计算机技术向各个领域渗透, 向世界每个角落延伸。

计算机硬件技术的发展推动着软件技术的发展, 使得功能丰富结构复杂的软件实现有了硬件基础; 软件技术的发展又刺激硬件技术的更新换代。人们在摩尔定律驱使下, 为软硬件

收稿日期: 1998-10-20

^{*}浙江省自然科学基金资助项目(397206)和“九五”浙江省科技攻关资助项目(961102160)

吴延熊, 男, 1967 年生, 助理研究员, 博士

升级而疲于奔命。微机操作系统从 Windows 3X, Windows 9X 到 Windows NT 都不断对运行环境提出更高的要求, 微机也相应地由 486, Pentium 提高到 Pentium II^[4]。

目前, 软件的发展比硬件的发展更为迫切, 更富有挑战性。在计算机问世之初, 由于受 CPU、硬盘和内存等硬件性能的限制, 只能遵循“指令尽可能精练”的原则, 人们更多关注的是算法改进和优化的技巧, 而不是软件的结构和界面。以操作系统为例, 剑桥大学开发的第 1 代 EDSAC 计算机运行的操作系统源程序只有 41 行, 堪称程序艺术之精品; 而到 1963~1966 年, IBM 开发 IBM360 操作系统时, 前后共投入 5 000 人, 整个操作系统的源程序量达到了 100 万行。

60 年代, 系统软件复杂化的同时, 应用软件也急剧膨胀。软件的规模、种类、功能和结构日益繁杂; 许多硬件尚未实现的功能, 需要软件加以补充; 各种不同的应用环境, 不能使用相同软件等因素造成的压力都集中到了软件开发人员的头上。然而当时软件开发人员大量匮乏, 技术水平不均, 加上手工作坊式的生产方式, 导致开发成本很高, 软件质量难以保证, 而且开发周期很长, 很难按期交货, 有的甚至半途而废。软件开发常常是一项相当原始的工作, 产生大量没有结构、资源使用效率低下、缺乏与其他程序真正互用的杂乱程序代码。这就是六七十年代出现的“软件危机”, 即软件成了阻碍计算机技术发展的“瓶颈”。

为了摆脱困境, 人们发展了 2 种关键技术: 面向对象技术和软件工程技术。基于这 2 种技术, 开发人员现在可以采用基于组件软件开发的新型软件开发技术, 即实现可靠的、小的对象的快速开发、装配和重复使用, 以生成复杂的程序。ActiveX 便是其中的精华所在, 它是团队开发对象互操作和共享式开发技术的基础。

2 ActiveX 与预警技术系统的开发工具

ActiveX 是由 Microsoft 在 1996 年正式命名的一项内容广泛的技术概念。它的理论基础是 Microsoft 的组件对象模型 (COM), 经历了 OLE, VBX 和 ActiveX 共 3 个发展阶段。ActiveX 已成为基于 Windows 平台的应用软件和工具软件的重要组成部分。

2.1 ActiveX 的基本原理

ActiveX 是一种编码和 API (application program interface) 的协议, 是一种基于组件开发的对象处理、分发和包装的技术标准。Active X 组件通常是动态链接库的形式, 因此必须在一个叫容器的独立执行软件中运行。这样的容器包括 Delphi, Visual Basic, Visual C++, Internet Explorer 和 Access 等等。

ActiveX 组件的数据输入和函数功能执行都必须通过容器, 因此 ActiveX 组件和容器都必须支持一些特定的接口协议。根据 Microsoft 相应的规格标准, ActiveX 组件应具备如下的性能机制。

属性和方法: ActiveX 组件必须提供属性的名称、方法的名称及参数, 通过这项机制容器可以存取和改变 ActiveX 组件的属性参数。

事件: ActiveX 组件由这项机制通知容器在 ActiveX 组件中发生的事件, 比如属性参数的改变, 用户按下鼠标左键等。

存储: 容器由这项机制通知 ActiveX 组件存储和提取有关信息数据等。

ActiveX 组件只要在 Windows 的 Registry 数据库中注册后, 就可以像其他 Windows 应用

程序一样发挥各自的功能。目前，已有很多厂商专门从事商品化的 ActiveX 组件开发，为开发预警技术系统提供了大量现存的 ActiveX 组件。它们可以实现诸如统计分析、图表显示、网络通信、硬件控制、系统维护等特殊功能。充分利用这些 ActiveX 组件，不仅可以减轻大量的开发任务，而且可以提高预警技术系统的水平。

ActiveX 是一个模块化的灵活对象，如果某个应用程序或网页需要增加一项特殊的功能，无须重写整个程序，只要灵活地插入一个具有此项功能的 ActiveX 单元即可实现。程序设计人员可以利用他所喜欢和擅长的程序设计语言，如 Delphi, PowerBuilder, Visual Basic, Visual C++，Jbuilder 和 C++ Builder 等来编写 ActiveX。特别是可以利用多种语言进行混合编程，生产较高效率的代码，为调用它的容器提供高速的计算、通讯和图像显示等功能。

ActiveX 的优势还在于它的动态可交互性，用户可以动态地在使用过程中，通过改变它的属性和参数，将一个一般用途的 ActiveX 定义成自己所需特定的 ActiveX。ActiveX 可以直接与操作系统进行交流，如通过文件系统进行文件读写等操作，还可能具备 TCP/IP 功能。如果用户在自己的应用程序中插入一个通信用的 ActiveX，它可在特定的 URL 处用 HTTP 协议访问一个网页，或按 FTP 协议下载一个远程文件等。

2.2 预警技术系统的开发工具

根据吴延熊提出的基于 C/S 体系结构开发工具的通用模型及其评价选型原则^[2]，选择 Delphi 作为预警技术系统的主要开发工具。Inprise 在正式定名 Delphi 之前，称它为 VBK (visual basic killer)，Delphi 采用 Object Pascal 为开发语言，并加入了可视化的开发环境。Delphi 原是古希腊一个城市的名字，古希腊人确信 Delphi 位于世界的中心。Inprise 期盼 Delphi 成为可视化开发工具的先驱与核心，期盼用户用一种全新的视角来看待 Delphi 的程序开发，这种视角便是面向对象技术的精华所在。Delphi 和 Pascal 的不同就如同 Pascal 和汇编语言之间的差异一样大。

Delphi 的思想源于 Microsoft 的 Visual Basic 的成功之道，而 Visual Basic 来自 Microsoft 的 Application Factory，这一面向对象开发环境的一个实现。Application Factory 代表 Microsoft 对未来程序设计环境的观点，这一观点无疑会左右未来程序设计的发展方向。这一环境对于开发者来说是一个有着丰富工具的工作台，其特性包括：生成对象的工具，处置对象的工具包括对象的组合、联结与浏览；可视化程序设计环境；支持面向对象和传统程序设计的外壳；可继承的、通用的应用构架和运行时实例程序等。

Application Factory 的核心是基于 Microsoft 称之为 GSC (global system cache) 的一种中心库。GSC 含有一个数据库，它存储着与给定软件项目相关的所有细节，如图 1 所示。

窗口环境的出现，实在是使用者的福音，因为它简化了传统的命令行下输入指令的方式，取而代之

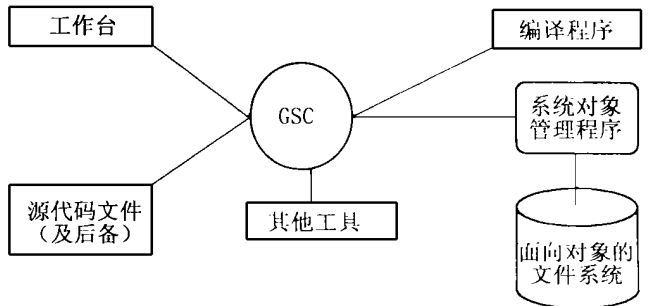


图 1 Microsoft 的 GSC 示意图

Figure 1 GSC demonstration diagram of Microsoft

的是更符合人性的、图形化的使用者界面。在图形化使用者界面之下,不必死记指令,借助更具弹性化的一个个窗口来表示一个应用程序,以鼠标来选择选项。但是对于程序设计师来说,却是一种很大的挑战^[3]。因为窗口的程序设计比传统的程序设计复杂许多。最初在 Windows 下开发窗口程序通常使用 C 语言配合 SDK,后来发展到了用 Visual C++ 配合 MFC 或者用 Borland C++ 配合 API,但是它们的复杂性和低效率也令人望而止步。因此,一个更为方便的软件开发环境——可视化开发工具应运而生。从 Microsoft 的 Visual Basic 到 Sybase 的 PowerBuilder,再到 Inprise 的 Delphi,一个比一个更为强大的软件工具简化了开发窗口软件的难度,并且提高了软件开发的效率。

Delphi 将面向对象技术、客户服务器结构、原代码编译器技术、阶段式数据库技术以及可视化开发环境融为一体。

在 Delphi 之前,软件开发分为 2 个不同的体系,其中之一是传统的编程语言如 C, Pascal 和 Basic 的世界,它包括这些语言和围绕它们的重点在于算术、数据结构和面向对象技术的产生。另一个则扩展到通用数据库管理软件包领域,如 dBase, Paradox 和 FoxBase 等。它产生初步的宏能力并经过一段时间后发展成为简单但实用的可编程命令解释器。在过去几十年里,2 个领域的发展都极为迅猛,但值得注意的是,二者没有一点混合渗透的迹象。现在 Delphi 将当今最传统的编程语言 Object Pascal 和强大的数据库技术融为一体。

2.3 预警技术系统开发工具的 ActiveX 技术

Delphi 完全支持 ActiveX 技术,而且可以很方便地创建、注册、安装、发布和使用 ActiveX 组件、ActiveForm 和 OLE 自动化对象。Delphi4.0 内建对 COM 的支持。由于 COM 对象具有语言无关性,因此用 Delphi4.0 创建的对象可以与 Visual Basic, Java, C++ 及其他语言实现对象的交互。在多层 C/S 环境中,COM 对象可以封装商业规则,为分布式客户提供服务。Delphi4.0 利用“Package (包)”技术来封装 ActiveX,以减少冗余的 VCL 框架代码,从而使应用程序更加精巧,更加适合于在 Internet/Intranet 上传输和分发。

3 ActiveX 在预警技术系统中的应用举例

目前,各类信息系统和应用程序的开发都会涉及 ActiveX 技术,特别是比较复杂的预警技术系统更离不开 ActiveX 技术。在开发县域森林资源预警技术系统 RFREWS 1.0 时,我们不仅创建了许多 ActiveX 新组件,而且充分利用了现有第三方开发的 ActiveX 组件。

3.1 ActiveX 组件的创建举例

Delphi 4.0 创建 ActiveX 组件,几乎达到了智能化的水平。它提供 2 个 ActiveX 创建向导:ActiveX Control 向导和 ActiveForm 向导。ActiveX Control 向导通过在 VCL 组件外包装一层 ActiveX 类,可以将现存的 VCL 或用户的 VCL 组件转化成 ActiveX 组件。ActiveForm 向导可以帮助程序设计人员从头设计一个新的 ActiveX 应用程序,向导建立一个工程,并提供一个空白表单,以便程序设计人员增加相应的组件来设计表单。

ActiveX Control 向导创建的应用单元,主要继承了 2 个对象:用于定义 ActiveX 组件的对象——TActiveXControl 和用于封装 VCL 组件的对象;ActiveForm 向导创建的应用单元,主要继承的对象是 TActiveForm。创建 ActiveX 新组件的步骤主要有以下几项。

3.1.1 设计和创建一个用户自定义的 VCL 组件,以便形成用户所需 ActiveX 新组件的基础。

3.1.2 利用 ActiveX Control 向导创建一个基于 VCL 组件的 ActiveX 新组件; 利用 Active-Form 向导创建一个基于 VCL 表单, 并可在 Web 上发布的 ActiveX 新组件。

3.1.3 利用 ActiveX 属性页应用向导创建组件的 1 个或多个属性页。

3.1.4 将属性页与 ActiveX 组件建立相应的关联。

3.1.5 ActiveX 组件的注册登记。

3.1.6 ActiveX 组件在所有的目标应用程序中进行测试。

3.1.7 通过 Web 网发布该 ActiveX 新组件。

3.2 ActiveX 组件的应用举例

Delphi 在默认状态下自身就提供了一些常用的 ActiveX 组件: Software FX 公司的 Chart FX, Pinnacle Publishing 公司的 Graph, Visual Components 公司的 Formula One, First Impression 和 VisualSpeller 等。当然, 用户可以非常方便地在 Delphi 中加载任何第三方 ActiveX 组件。这些 ActiveX 组件就成了预警技术系统的基本构件, 为研建功能强大的技术系统提供了基本素材。我们研制县域森林资源预警技术系统 RFREWS1.0 时, ActiveX 组件技术。

Software FX 公司的 Chart FX 组件在预警技术系统图表管理器开发中的应用。

3.2.1 Chart FX 组件的属性主要有: About, PaletteBar, SerLeg, Value, Adm, IniValue, Pattern, ShowDialog, VertGridGap, Decimals, ItemColor, PatternBar, ShowStatus, View3D, Angles3D, DecimalsNum, ItemWidth, PixFactor, Stacked, ViewRot3D, AutoIncrement, Edit, ItemStyle, PointType, WallWidth, BarHorzGap, KeyLeg, StatusText, BGapFixedBkColor, KeySer, Style, XlegType, BkColor, FixedColor, ReturnValue, Tag, Xvalue, Chart3D, FixedGap, Legend, Rgap, TopGap, Yleg, ChartStatus, FixedStyle, LegendWidth, RGB2DBk, ThisBkColor, ChartType, FixedWidth, LeftGap, RGB3DBk, ThisColor, FixLeg, LineBkColor, RGBBarHorz, ThisPoint, Color, Fonts, LineColor, RGBBk, ThisSerie, Const, LineStyle, RGBFontThisValue, ConstType, Grid, LineWidth, Title, hCtlWnd, Nseries, hFont, Nvalues, Scheme, ToolBar, CustTool, hText 和 Type 等。

3.2.2 Chart FX 组件的方法主要有: CloseData, CopyData, CopyBitmap, DblClk, ExportFile, ImportFile, OpenDataEx, Language, Paint, PrintIt, ReadTemplate, RigClk, Scroll, SetStatusItem, SetStripe 和 WriteTemplate 等。

3.2.3 Chart FX 组件的事件主要有: ChangeColor, ChangeValue, RbuttonDown, ChangeFont, Destroy, ReadFile, ChangePalette, GetLegend, ReadTemplate, ChangePattern, GotFocus, UserScroll, ChangePatPal, LbuttonDblClk, ChangeString, LostFocus, ChangeType 和 Menu 等。

3.2.4 Chart FX 组件的应用: , , Chart FX 组件, , 2 所示的图表管理器。

、 、 、 HILOW 图、RARETO 图等 10 余种二维三维统计图。

4 结论与讨论

我们研制县域森林资源预警技术系统 RFREWS 1.0 时, Delphi, ActiveX 技术

； 2 项与 ActiveX 相似，
 ， CORBA 和 Java 技
 术。

4.1 CORBA 和 COM: 2 种相互竞争的标准

组件软件开发的优势在于可重复使用、

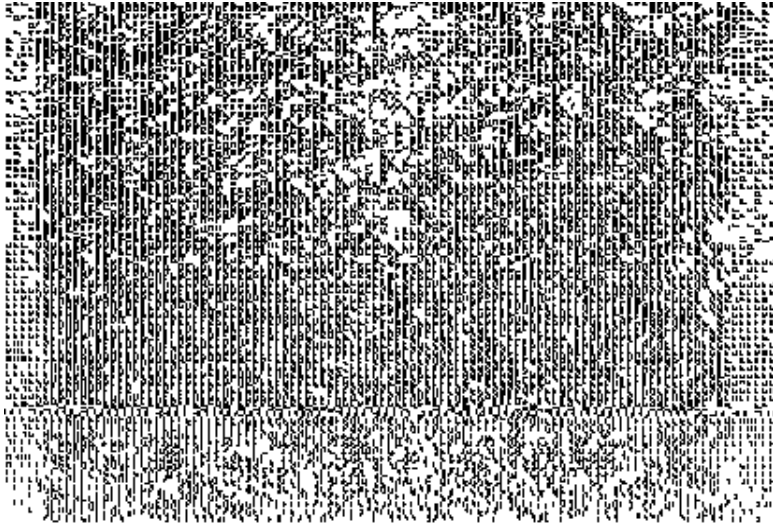


图 2 预警技术系统的图表管理器

Figure 2 Chart management of early-warning technical system

件的用户和开发人员来说，
 2 个阵营：Microsoft 的组件对象模型（component object model 即 COM）（common object request broker architecture 即 CORBA）。ActiveX 以 COM 为基础， COM， COM 技术的集中体现。

2 个标准的目标都是让开发人员生成带有可预测的标准接口的组件。

CORBA 是跨行业的可以通过网络分配下发的标准组件的技术规范。 CORBA，

（object request broker 即 ORB）

。CORBA 提供 4 种主要的组件：ORB、

COM 也具有相似的功能， Windows 的组件对象模型。

， CORBA 和 COM 都在迅速发展， Object Management Group 于 1998 年 9 月推出了新版 CORBA 3.0， Microsoft 也同期推出了 COM+。

CORBA 和 COM 协调一致地运行。 Delphi 4.0 能同时支持 CORBA 和 COM， In-

prise 还制造用于其面向 CORBA 的 Visibroker 的 COM 桥, COM 的 Microsoft 竞争产品 VB, VC 等具有更广泛的应用范围。

4.2 ActiveX 和 Java Beans: 2 种相互对抗的技术

Java 和 ActiveX 一样也是一项广泛的技术概念, www 的 Java Applets 和近年发展起来可动态调用的 Java Beans 都是与 ActiveX 组件相类似的模块化程序单元,

Java 这个编程语言写成。

ActiveX 通常只限使用于 Windows 9X 和 NT 操作系统, Macintosh 虽也可支持 ActiveX, 。Java 则与之不同。Java 最大的好处是与各种不同的操作系统兼容, “ , ”, Java 虚拟机 (JVM) 。JVM 解释 Java 编译后的 bytecode, , Java 本身可全然不关心具体操作系统的细节。Java Bean 利用 “ ” ,

ActiveX。ActiveX 和 Java Bean, ? ,

。CORBA 和 COM 一样, ActiveX 和 Java Bean 也出现了相互利用的 “ ” , Java 的开发者 JavaSoft 已在 Beans Development Kit (BDK) ActiveX Bridge, IBM 在 VisualAge 中引入了 Bean Migration Assistant。

- 1 . Windows 95 实用工具参考大全. : , 1998. 1~10
- 2 . : []. : , 1998
- 3 Capers J. *Applied software measurement: assuring productivity and quality*. New York: McGraw-Hill Inc, 1991. 1~107

Wu Yanxiong (Key Laboratory of Yunnan Academy of Forest Sciences, Kunming 650204, PRC), Zhou Guomo, and Guo Renjian. ActiveX and its application in regional forest resources early warning technical system. *Journal of Zhejiang Forestry College*, 1999, 16 (1): 100~106

Abstract: The paper discusses the important development technology of regional forest resources early-warning technical system, which is ActiveX technique. From the relationship between ActiveX and software crisis, the importance of ActiveX control technique is discussed. The basic principle of ActiveX control technique and the ActiveX supporting of development tools are briefly introduced. Finally, the paper gives examples of ActiveX control creation and application which occurred in early-warning technical system development, and compares the ActiveX with CORBA and Java Bean which are the same technique with the ActiveX.

Key words: regions; forest reserves; early-warning technical system; Delphi 4.0; ActiveX control; application