

汽修 MIS(网络)实现中的技术处理

王永刚 (河南大学计算机应用技术研究)

摘要:本文总结了汽修 MIS(ARMIS)实现中的一些技术要点,阐述了在 WFW 平台上进行 MIS 开发的方法。

一、汽修 MIS(ARMIS)简介

ARMIS(Automobile Repaired Management Information System)是一个面向汽修行业的计算机网络管理信息系统。系统由汽修生产管理、配件经营管理、财务管理、技术管理和综合事务管理等若干个子系统组成。它能够对维修的车辆、项目、材料、生产人员、设备、资金和帐目等实施全面的管理,有效地解决了该行业中普遍存在的核算难、管理规范差等深层次的管理问题。该系统采用了客户/服务器模式,选用 WFW(Windows For Workgroups)作为开发平台,以 FoxPro 为主,集成了 C、VisualBasic、汇编语言等多种开发工具,具有良好的开放性、可扩充性、可维护性、可升级性和较高的安全性。本文将对 ARMIS 开发中的几个关键性技术问题进行阐述。

二、合理选择系统开发平台

ARMIS 之所以选取 WFW 作为系统开发平台,主要出于两方面的考虑:一方面,它能够较好地体现面向对象的方法和客户/服务器计算模型,具有较强的软件集成能力;另一方面,WFW 具有强有力的分布式处理功能,适合于汽修行业管理的特点。该系统以汽修企业中各职能部门为独立的网络结点,建立分布式的网络系统。考虑到企业内部各个部门的业务量相差悬殊,对业务量较大的部门,可将网络结点扩充成为具有客户/服务器结构的子网,下挂于对等结构的分布式网上,成为一种混合型结构的对等网络。

这种网络结构使系统具有如下特点:

首先,按部门分布存放数据不仅增加了数据的安全性和可靠性,减轻了网络负担,而且保持了数据的相对完整性,便于数据管理。

其次,功能的分布处理可更好地适应用户的业务需要,在更高层次上满足用户对数据保密性的要求;同时可有效地利用硬件资源,提高系统的运行效率。在同样的

性能指标下,降低了系统成本。

第三,客户/服务器模式的计算,不仅进一步减轻了网络负担,增加了数据访问的安全性(服务器只接受已授权用户的访问),而且保证了系统良好的开放性。

第四,具有良好的伸缩性。系统规模可随企业规模的大小进行适度调整,既可根据企业下属部门的多少增减上层网络结点个数,也可根据各部门业务量的大小对下层子网进行放缩,或者把下层子网从上层网中独立出去,作为一个独立系统运行。

第五,易于将网络环境升级到 Windows NT 或 Windows 95。因为它们均内置了 WFW 的点对点网络功能,对 WFW 具有良好的兼容性。

三、与汽修行业相关的几个具体问题的解决

核算困难,尤其是单车成本核算、劳动部门核算和生产人员劳动效益核算更困难,这是汽修行业管理中最为突出的问题。

及时准确地跟踪汽车修理过程中项目和材料的不断变化,是堵塞管理漏洞、准确把握企业运行状况所必须解决的关键性问题。

针对汽修行业中普遍存在的这些问题,我们在数据结构设计、数据处理方法的选取、查询方案的确定等方面均给出了有效的解决方案。

1. 具有复杂对应关系的数据处理

由于汽车修理过程中存在着许多有复杂对应关系的数据,如生产系统中对车辆修理的动态跟踪和经营系统中的拆散销售就是典型的例子。一辆车的修理可能要修理多个项目,由多个部门的不同人员来承担,修理一个项目要用多种材料,一种材料又有多种规格和型号等,形成了错综复杂的对应关系。因此,在系统中必须妥善而有效地处理这些具有一对多或多对多等复杂对应关系的数据,对此我们采用了关键字查找、多库关联和 MEMO 字

段技术等方法加以解决,取得了良好的效果。

2. 动态演变性数据的处理

在汽车修理的过程中,每辆车的修理项目、所用材料随着修理的进展而不断发生变化,它的生产计划调度环也处于不断的循环之中,每次循环均会使数据发生演变。“有限的计划-调度环”较好地概括了这种动态演变性。

例如,对于一个正在修理中的任务,以该任务的初始维修计划作为迭代的初值 P_0 ,以后每进行一次循环,均使得该任务的维修计划从 P_i 变化到 P_{i+1}

$$P_{i+1} = f(P_i, \lambda)$$

其中 λ 是引起 P_i 变化的项目和材料,在 P_i 中以状态标记区分计划中的项目和材料是否生效。

3. 零件查询的快速定位

从几万种甚至几十万种零件中快速挑选出某一所需的具体零件,是对系统实时性的要求,也是影响系统效率的关键因素。为此我们通过建立零件的特征检索树,采用最短搜索路径快速定位法,取得了比较理想的效果。

四、有效地使用 WFW 的网络功能

1. 网上数据交换

WFW 提供了两种基本的数据交换方式:文件共享和网络 DDE。

WFW 中各网络结点以文件共享的方式交换数据时,最关键的是解决同共享数据的连接与断开问题。共享数据的提供通过 WFW 的 File Manager 来完成。数据共享方式可分为只读共享、全共享(可写数据)和依据口令权限提供只读与可写共享三种方式。共享数据的连接在 WFW 中也必须通过 File Manager 来完成。但这种连接一方面无法摆脱手工干预,不能在程序中使用;另一方面,也不能通过设置 WFW,使其开机自动连接。如果这样,一旦两台机器分别需要连接对方,开机时将总有一台机器连接失败(因为两台机器不可能绝对的同时进入,因此必然有一方因进入系统稍早而无法连接成功)。我们通过一系列的试探性工作并查阅了 DOS 和网络方面的大量资料,最后用 C 语言构造了一个动态连接库(DLL),通过调用 DOS 功能实现了共享数据的实时连接和断开,并可根据用户的权限允许其以只读方式或可写方式访问

共享数据,较好地解决了数据共享的问题。

DDE 在 Windows 2. X 版中即得到支持,而网络 DDE 则是在 WFW 中首次提出,之后又在 Windows NT 和 Windows 95 中被不断扩充和发展。WFW 对网络 DDE 的支持使其应用程序的良好开放性被扩充到了整个网络中,但如何正确地使用网络 DDE 完成数据交换却不是一件容易的事。在 Microsoft 提供的有关 WFW 的技术资料中并未谈及网络 DDE 的用法,也未见到 WFW 的 SDK。虽然在《微软专刊》(试刊号)等杂志上谈到网络 DDE 的部分用法,但因其说法错误,无法使用。我们通过对单机 DDE 的深入分析和大量实验,成功地解决了网络环境下 DDE 的应用问题,并订正了使用网络 DDE 的一些错误提法。ARMIS 中生产系统向经营系统进行修理用料的价格查询和各个子系统所使用的基础编码库的同步维护等功能都是通过网络 DDE 实现的。

2. 网络状态检测

分布式的对等网络结构,使得网上各站互为客户端/服务器,不存在专用服务器,因此在其上开发的系统应具备自动检测网络互连的功能,以确保网络的运行安全。一方面客户端试图与服务器连接时,发生连接失败,应能自动返回;另一方面,当服务器退出服务时要能够检测到有无客户端正连接着自己。如果一台正在提供服务的机器突然下网,不仅会造成客户端数据的丢失,而且可能引起系统错误,而 WFW 在这种情况下,虽然有警告但依然允许操作员强制使服务器下网。为防止这种情况的发生,我们用 C 语言构造了一个动态连接库完成上述检测功能,并在系统程序退出时进行检查,一旦发现还有客户端使用本机资源时,则阻止系统退出直至所有被使用资源均被释放,确保了网络和系统的安全。

3. 网络消息传递

在网络中的各个结点间进行消息传递是网络的基本功能之一,在 WFW 中,具有这种功能的程序有 Mail, Chat 和 WinPopUp。但这些程序使用起来有诸多不便:如 Chat 不能传送汉字信息,Mail 不仅对操作员的要求高而且必须开通网上邮局才可传递信息,WinPopUp 则只在 WFW 3.11 版本中才有;如将它们包装到应用系统中不仅使系统显得笨重、影响运行速度且不能做到无缝衔接。为此,我们基于网络 DDE 重新构造了一套消息传递的方法,使用快捷方便,深受用户的喜爱。

五、保证数据安全的几项措施

数据的安全性是系统设计中须重点考虑的问题,对管理信息系统尤其重要。因此为确保系统的数据安全,我们在具体实现时采取了如下几项措施:

1. 将数据资源按权限划分

在系统中把数据按部门存放,各部门的数据分别存放在各自的工作站上。每个部门的数据均被分为私有和公共(共享)的,分别存放于不同访问权限的子目录之下。对于共享数据则依据访问口令级别提供只读或可写权限。同时在用户进入系统时授予其一定的操作权限,不同的用户只能操作授权其使用功能中的数据。

2. 数据完整性检查与保证

每次进入系统时,首先检查系统所用到的数据库、索引文件和其它文件是否存在、是否合法、是否完整。检查时若发现文件丢失、异常或不完整,系统则自动进行有限度的数据修复或告知操作员如何进行修复并中止系统运行,以避免对系统和数据造成更大的破坏,保证了系统的健壮性。

3. 通过功能跟踪监视系统运行

对系统中一些重要的操作,设置了系统自动跟踪功能。一旦系统运行错误或异常,则可根据跟踪记录去恢复有关数据,保持数据的完整性。

4. 完善的数据备份与恢复功能

每个工作站上的数据按其变化的频度可被分为三类:当前发生数据、临时运行数据、历史数据等,各个子系统均可对本地的数据进行增量备份或全部备份,并可随时恢复备份的数据,确保系统运行过程中的数据不会发生丢失。同时,对一些关键的数据库,采用即时备份的方式,当数据库内容发生变化时,立即进行备份。

5. 使用数据库加密技术

采用通用的数据库开发平台进行系统开发时,数据库的安全是个突出的问题。如果对之不加以保护,则系统的运行数据可被他人方便的篡改、企业的机密数据也容易泄密。对此我们采用C语言编写了一个动态连接库完成数据库的加、解密工作。在系统进入时将数据库解

密,而在系统退出时再将数据库重新加密存放,确保数据内容的安全。

六、辅助开发的几项工具

在系统的开发过程中,由于开发工具本身的不足与限制,导致了系统程序的开发和文档资料的整理繁琐、不便。为此我们制作了一些实用的辅助工具,它们为数据字典的维护、程序员的编程以及技术文档资料的整理提供了方便,确保了面向对象原型开发方法的顺利实施。

1. 数据字典的维护工具

采用面向对象的原型法进行系统开发时,往往需要经过若干个回合才能把系统模型逐步完善。而每一个回合都可能引起对象的属性和方法的局部改变(尽管确定系统初始模型时需要尽可能考虑完整,但修改是不可避免的)。在RDBMS中,对象的属性被映射到Table的表列上。属性的变化将会引起表列的变化,即数据字典的内容发生了变化。怎样保持数据字典的同步修改是件非常麻烦的事情,因此我们制作了一个自动维护数据字典的软件工具。一方面,程序员可以通过它建立与维护数据字典,并由数据字典产生和修改数据库结构;另一方面它能够自动地搜寻系统各级目录(及其子目录)下的所有数据库,将其结构反馈回数据字典中,并将结构相同的数据库在数据字典中进行合并,且能根据已有数据库结构的注释自动地对未加注释的字段进行注释。

2. 代码片段的搜集与回送

FoxPro为程序员提供了功能强大的屏幕生成器和菜单生成器,程序员通过定义对象的指定代码片段去描述对象的行为,这为系统界面设计和编写事件驱动的程序带来了极大的方便。但是当程序员需要对多个对象的代码片段或同一对象的不同代码片段同时进行修改时,将会感到很不方便—需要不停的查找和切换编辑窗口。为此我们制作了一个工具,它既能方便地搜集屏幕生成器和菜单生成器中的程序代码片段,在逻辑上合并起来供程序员修改,也能将修改过的代码片段再回送到屏幕和菜单的相应对象中,同时还可将FoxPro生成器生成的含混不清的过程和函数名转变为清晰明白的名称,使生成后的程序便于阅读和整理。不仅方便了程序的修改,提高了编程效率;而且为系统程序文档的整理提供了方便。