

文档库

Notes

数据库模型
数据库管理

20

Lotus Notes 文档库分析

Analysis of Lotus Notes' Document Database

80-84

陈海广 李磊

G 276

(中山大学岭南(大学)学院软件所 广州510275)

摘要 This paper analyzes the groupware (Lotus Notes) from perspective of DBMS. Focus is set on the capability of entity-relationship description and manipulation of both data and relation in Lotus Notes' document data model. A comparison between Lotus Notes' data model and the traditional data models is also presented. Finally, the principle of applying Lotus Notes is proposed based on the analysis above.

关键词 Database management system, Document database, Groupware

近年来,飞速发展的信息科学技术,尤其是 Internet/Intranet 信息技术,在现代企事业和工商行政机构管理中逐渐扮演越来越重要的角色。由美国 Lotus 公司推出的 Lotus Notes 群件是这其中一个特色鲜明,具有代表性的应用软件。它最早系统地引入了 workflow 概念,以共享式文档库为基础,以电子消息通信为流程实施的基本手段,是目前最完善地实现实际工作流程的应用软件。Lotus Notes 在我国正引起越来越广泛的关注,被越来越多的大型企事业单位和行政机关所采用。但是,由于缺乏对 Lotus Notes 这一软件产品的全面深入了解,可能导致在利用 Lotus Notes 开发不同的应用系统过程中,出现截然不同的开发效率,产生截然不同的应用效果。本文将论述 Lotus Notes 文档库特性,对文档库数据模式和数据模型进行分析,对数据模型中描述实

实施。我们认为具有用户界面友好、系统易实现的特点。整个系统用 Turbo C 在 486 微机上实现,源程序达 50000 余行,已通过鉴定。目前,正在进一步研究关系的多继承与传递继承以及嵌套关系的索引机制等。

参考文献

- [1] Schek H. J. and Scholl M. H., The Relational Model with Relation-valued Attributes, Information Systems, 11(2), 1986
- [2] Dadam P. et al., A DBMS Prototype to Support Ex-

体联系的能力和对联系的操作进行横向分析和纵向比较,力图从文档库数据模型角度提出什么情况下适宜利用 Lotus Notes 进行开发,而什么情况下不适合的一般原则。

Lotus Notes 文档库数据模式

Lotus Notes 文档库基本上符合数据库的三级模式结构——即用户模式、逻辑模式、存储模式。但它与传统数据库相比,又有很多不同点。下面着重讨论其逻辑模式。

一、Lotus Notes 文档库逻辑模式

1. 数据模型

一个 Lotus Notes 文档库中不仅包含文档数据,而且包含文档结构定义(数据字典)、用户视图、存取控制表、程序代理等,这是目前许多流行数据库

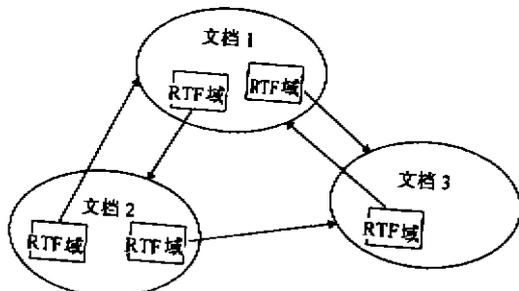
tended NF² Relations: An Integrated View on Flat Table and Hierarchies, In: ACM SIGMOD, 1986

- [3] Linnemann V., Non First Normal Form Relations and Recursive Queries, An SQL-Based Approach, In: 3th Intl. Conf. on Data Engineering, 1987
- [4] Stonebrake M. et al., Extending a Database System with Procedures, ACM Trans. on Database Systems, 12 (3) 1987
- [5] Lawrence A. R. and Stonebrake M., The POSTGRES Data Model, In: Proc. of the 13th VLDB Conf., 1987

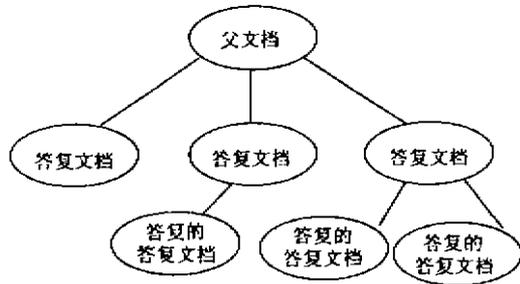
产品常用的做法。文档库中的一个数据记录为一个文档,一个文档可以包含若干个“域”(Field),这是文档库中不可分割的最小的数据单位。文档结构、域的属性等由数据字典——Lotus Notes 中称为“表单”(Form)定义;在表单中不仅可以定义一系列域的集合,还可以同时定义文档显示打印的格式。一个文档库中可以包含多个表单,定义多个文档格式。

文档库的“域”可以有数字、日期、时间、文本等常规的数据类型,此外还支持一类特殊的域类型:RTF 类型。这一类型的域支持 Richtext 数据格式,可以包含多媒体数据,支持 DDE/OLE 对象嵌入和链接,以及由其他应用软件产生的数据,如 Excel、Powerpoint、Word 等。

Lotus Notes 文档库中为记录之间建立关系(即文档之间的联系)提供两种方式。一种是通过 RTF 类型的域建立与另外文档的链接,例如用户可以在邮件文档中使用 RTF 文档链接指向另一个共享的文档,使所有收信人与发信人都看到同一个文档,并都能随时了解这一文档发生的变化。这种文档之间的链接是单向的,文档中可包含的链接个数、文档自身被链接的次数、链接文档的属性都没有限制,是一种网状的关系,但所有的 RTF 链接域必须要预先在表单中定义好,另外一种为建立父(主)文档-答复文档或答复文档-答复的答复文档关系,首先在表单中定义一类文档为答复文档或答复的答复文档,然后指定具体答复文档所对应的父文档或者指定具体答复的答复文档所对应的答复文档。每个父文档可以有多个答复文档,每个答复文档也可以有多个答复的答复文档,但每个答复文档只可以有一个父文档,每个答复的答复文档也只可以对应一个答复文档。这是一种层次型的关系,两种文档(记录)关系见下图:



Lotus Notes 文档间通过 RTF 域建立链接



Lotus Notes 文档间建立主-从关系

2. 数据完整性和安全性

Lotus Notes 在文档表单定义中以多种方式支持数据完整性与安全性约束。用户可以在完整性约束列表中直接选取常用的预先定义好的完整性约束,也可以利用 Lotus Notes 提供的公式语言(Formula)或脚本语言(Lotus Script)自行定义完整性约束。安全性约束的选取或定义与完整性约束基本一致。在 Lotus Notes 中安全性约束可以针对文档库、文档、域实施,最常用的方法是采用存取控制表,管理员定义好认证服务器和组、角色类别、用户等的权限,在数据库设计中可以将这些权限自由地组合并赋予相应的文档库、文档或域。Lotus Notes 还可对文档数据提供加密和电子签名,其密钥长度在北美发行的版本中为 64 位,其他地区发行的版本中为 40 位。

3. 对文档数据和文档关系的操作

Lotus Notes 提供多种方式对文档数据及文档间关系进行操作。最简单的方式是在用户模式下定义一系列文档视图,按一定关系列出文档,然后直接在视图上执行文档浏览、删除、插入,以及建立、修改或取消父(主)文档-答复文档或答复文档-答复的答复文档关系。用户按表单定义的格式浏览或修改文档域上的值,建立、修改或取消 RTF 链接关系,并可以选择同时浏览或修改与该文档相关联的另一个文档,如 RTF 链接文档、父文档等。

Lotus Notes 还可以通过内置的公式语言或脚本语言对数据和关系进行操作。公式语言类似于宏语言,提供许多被称为“公式”的内部函数和简单的程序流控制机制,可用于编写常规的较简单的应用。Lotus Script 是一种面向对象的功能强大的编程语言,其语法结构和编写方式都类似于 Visual Basic,

Lotus Notes 中的数据库、文档、域、视图以及对它们都被封装到相应的类中。这些类对 RTF 链接关系提供了一些简单操作,如在给定 RTF 域上建立链接、断开链接和获取链接文档;对父(主)文档-答复文档或答复文档-答复的答复文档关系则提供了较完整较完整的操作,除了建立与断开联系外,还提供了为当前文档获取上一级文档(父文档或答复文档)、获取下一级文档(答复文档或答复的答复文档)、获取同级兄弟文档等操作。在 Lotus Notes 中用公式语言或脚本语言编写的程序以图形化按钮操作、代理、或表单和域上的事件操作等几种方式运行。

对文档数据的操作除了以上方式外,还可通过第三方软件开发商为 Lotus Notes 开发的应用系统或 Lotus Notes 本身的 API 接口直接操作文档数据。

4. 文档库定义

Lotus Notes 采用可视化设计和对话框人机对话方式来定义文档库,实现了“所见即所得”。用户可直接在计算机屏幕上如同平面设计般设计表单格式,例如插入或删除域、表格、固定文字、图片声音等,并通过拖动布置它们的位置,通过属性对话框可以交互地设置表单底色、域类型、域显示格式、域显示所用字体字号、域的完整性安全性约束、文档类型(一般文档、答复文档或答复的答复文档)、文档存取控制等众多的属性,还可以编写有关文档、域的各种事件操作或添加用户自定义的按钮操作。Lotus Notes 文档库定义工作完全可以由上述方法完成,不再依赖传统数据库中的模式描述语言 DDL 来书写数据库基本结构和内容,这样可使那些对计算机不熟悉、未受过专门培训的人员在较短时间内根据需求很快设计出完整、实用的文档库,并且有利于日后修改调整,以适应新的变化。

Lotus Notes 文档库逻辑模式-物理存储模式的映象由系统自动实现,并且被系统完全掩盖,用户在不同硬件和操作系统平台上都可获得统一的逻辑模式。

二、Lotus Notes 文档库用户模式

Lotus Notes 文档库用户模式由在表单中定义的文档显示格式、用户视图、导航器以及代理程序等组成。

前面已经指出,定义文档的表单不仅定义了组成文档的各个域,还定义了文档显示打印的格式,甚至可以定义单选列表、复选列表、按钮操作、热点操作(HotPot)等标准的窗口系统控件。通过表单定义中的存取控制,可以使同一文档以不同的形式呈现在不同的用户面前。

Lotus Notes 中的用户视图是一个二维的表格,视图中的每一行代表文档库中的一个文档或某一组文档的统计值,各个列则可以显示文档中某个域的值或与某几个域相关的值,也可以仅是文档编号等与文档域值无关的内容。用户视图中可仅显示对文档库执行了选择、投影或排序等操作所得的部分文档和文档中的部分内容;文档显示可按一定的层次排列,这个层次或者是经排序分类形成的,或者是由父(主)文档-答复文档或答复文档-答复的答复文档这种层次关系形成的。但用户视图中无法显式地体现 RTF 域的文档链接关系。

Lotus Notes 中用户视图的定义方法类似于文档的表单定义方法,也利用了可视化设计和对话框人机对话方式,实现“所见即所得”。用户可以任意添加、删除各个列,并随意调整列的宽窄和顺序,在属性对话框中可以交互地设置各列的内容、格式、排序方式等,还可以编写有关视图、列的各种事件操作或添加用户自定义的按钮操作。视图的定义也不依赖传统数据库中的外模式描述语言 DDL,便于一般用户掌握和经常性的修改调整。

在 Lotus Notes 中也可以利用公式语言、Script 语言或 Notes 本身提供的固有的简单操作来编写一类特殊的应用程序——代理。编写代理时必须指明代理处理的数据对象;所有文档、视图、或视图中选定文档等。通过执行代理程序可以对文档进行成批处理。代理的执行可以是手动的,也可以由事件触发自动执行或定时执行,这些对于流程的实现和控制是必不可少的。

导航器是 Lotus Notes 中一类特殊的用户界面。程序员可以在导航器中定义形象直观的按钮、图形热点等,从而建立与实际不同文档、视图或代理的联系,最大限度地简化最终用户的操作。导航器的定义类似于视图的定义,这里不再赘述。

用户模式-逻辑模式的映象由 DBA 在视图、导

航器、代理的定义过程中自行定义并维护。

三、Lotus Notes 文档库物理存储模式

Lotus Notes 在底层采用了 X.500 目录管理工业标准,详情请参阅有关资料。

四、Lotus Notes 文档库数据模型与网状、层次、关系数据模型比较

由前面对 Lotus Notes 文档库数据模型的分析可以看出,它与网状数据模型有某种程度上的相似之处:Lotus Notes 中通过 RTF 域建立的文档之间链接可以使一个文档有多于一个的双亲文档,当然也可以一个双亲文档也没有。但实际上二者有很大的区别。以网状数据库的典型代表 DBTG 系统为例。DBTG 系统中有一个很重要的描述记录间一对多关系的概念——系。在 DBTG 系统中除了定义记录的完整性约束以外,也要对系定义完整性约束;除了提供对记录的操作,还提供完整的对系的操作。反观 Lotus Notes 中文档间的 RTF 链接,则不存在与系相对应的概念。在 Lotus Notes 中用做文档链接的 RTF 域必须在表单中预先定义好,并且不能在表单定义以外的地方动态地增加和删改;Lotus Notes 允许在编辑文档时建立、修改或删除 RTF 域文档链接,但这要求用户自己动手完成,并且需要用户自行选择被链接文档。Lotus Notes 开发工具提供了对链接的操作,但只有在给定域上建立/断开链接和获取链接文档等有限的几个,不能象 DBTG 系统那样提供对系值的操作。理论上在 Lotus Notes 中可以利用文档间的 RTF 域链接实现 DBTG 中系的概念和对系的操作,但编程人员要针对具体应用自行模拟系的逻辑结构和编制对系值的操作,这不仅要求编程人员有很高的编程技巧,也破坏了数据的独立性,并且在文档库数据大时程序运行效率极低。

Lotus Notes 提供的另一种建立文档间联系的方式是建立父(主)文档-答复文档或答复文档-答复的答复文档关系;每个父文档可以有多个答复文档,每个答复文档也可以有多个答复的答复文档;但每个答复文档只可以有一个父文档,每个答复的答复文档也只可以对应一个答复文档,可见这种文档关系可以与层次数据模型相对应, Lotus Notes 允许在视图中建立、修改或删除文档的这种层次关系,但要求用户自己动手完成,并且需要用户自行选择上一

级文档, Lotus Notes 开发工具对这种层次关系提供了一系列较完整的操作,如获取当前文档的上一级文档(父文档或答复文档)、获取当前文档的下一级文档(答复文档或答复的答复文档)、获取与当前文档具有相同上级文档的同级兄弟文档等,这些操作与典型的层次数据库系统 IMS 对数据片段(一个数据片段相当于一个记录)关系的操作是极相似的。但 Lotus Notes 中这种文档的层次关系最多只允许两层,大大限制了其应用的范围。层次模型本身具有固有的不足,现实世界中实体的联系常常呈现网状而非层次型,因而层次模型对实体联系描述能力较弱,并且常迫使数据冗余出现,在 IMS 中为了补救这一点,引入了逻辑数据库的概念,利用数据片段中的逻辑指引元将网状结构改造成新的层状结构。在 Lotus Notes 中也可利用文档间的 RTF 链接域作逻辑指引元,将网状结构改造成层状结构,但由于 Lotus Notes 的层次关系只有两层,使得即使引入 RTF 链接域,其描述记录关系的能力仍然不足。

Lotus Notes 文档库数据模型与关系模型相比,则基本没有共同之处。在关系模型中,无论实体还是关系都由统一的逻辑结构——二维表描述。而在 Lotus Notes 文档库数据模型中则没有二维表的概念,自然也就不提供象 SQL 这一类标准的关系数据库操作语言。现在有一些第三方软件开发商推出了可以在 SQL 与 Lotus Notes 全文检索或关键字检索之间高效地相互转化的软件产品,这些产品涉及复杂的算法和 Lotus Notes 底层的实现技术,非一般应用软件开发人员可以做得到。新版本的 Lotus Notes 有可能将关系操作集成到 Lotus Notes 内部,这将大大改善 Lotus Notes 数据模型对数据关联的处理能力。

由以上分析可见,在有些关于 Lotus Notes 的文章中将其描绘成是一个管理“关系”的系统,实际上 Lotus Notes 擅长管理的是数据和不同部门、不同组织成员之间发生的关系,这种关系体现在所谓的工作流的不同阶段上,是数据与外部的联系; Lotus Notes 并不擅长处理数据之间的关系,这种关系是数据内部固有的,并且现实世界中存在着大量的这种数据间的内部关联。

用 Lotus Notes 开发一些应用系统时效果不

好,其中一个重要原因就是某些应用系统中存在大量复杂的数据内部关联,以图书管理系统为例,我们在将实体关系向 Lotus Notes 文档数据模型转化时,遇到了极大困难。在图书管理系统中有几种基本的实体数据信息,如书种信息、每种书的书册/分册信息、每册书的借阅情况,每册书的著者、出版社和典藏信息,以及读者借阅情况等。这些信息之间有密切关联,在关系模型中可以有清晰的函数依赖关系,但在 Lotus Notes 中无法使用关系数据模型,也很难利用 RTF 域链接建立网状模型(因为仅书种信息就有超过8万条记录),最后只能勉强借用父(主)文档-答复文档和答复文档-答复的答复文档这个两层关系建立模型,书种信息作为主文档,书册/分册信息作为书种文档的答复文档,而每册书的每次借阅信息作为书册/分册文档的答复——即作为答复的答复文档;读者借阅信息通过 RTF 域链接相应的书册/分册文档;每册书的著者、出版社和典藏信息等数据冗余度较大,本应单独为它们建立文档,但由于要求视图中连接显示这些信息和书册/分册信息,而在 Lotus Notes 中很难处理连接关系运算,并且也无法动态地处理由著者、出版社或典藏文档到书册/分册文档的连接关系,故只有将这些信息并入书册/分册文档,增加数据冗余。最后开发出来的这个系统只能处理上述3类实体信息的动态关联,功能很少,并且由于数据冗余使文档库极其庞大,搜索查询的效率极低,库中存放200条书种文档、不超过500条书册/分册文档,在 Pentium 133/32Mb Ram 机器上运行,其查询速度已明显太慢。

结束语 Lotus Notes 是基于文档库的面向工作流的群件,但并不是任何 workflow 都适宜用 Lotus Notes 来开发。Lotus Notes 适用于这样的工作流: workflow 中的数据内部关联很少并且很简单,数据量较小并且非结构化,要求支持多媒体文档和长度变化较大的文档,要求实时性不高,要求可以根据实际情况的变化很快很容易地调整、修改原有数据库和流程等。Lotus Notes 不适用于这样的工作流:数据量大、结构化程度高,实时性要求高,数据内部关联繁多而复

杂, workflow 比较固定,没有多大自由发挥的余地。这一类 workflow 系统一般是核心业务系统,仍应采用大型关系型 DBMS 来开发。另外还有一类 workflow 应用系统,要求根据新的需求对原有的非常庞大和复杂的各部门、各组织、各成员之间分工合作的流程进行改造,这一类应用属于目前国外风行的 BPR (Business Process Reengineering)——企业过程再工程的范畴,它往往是一项庞大的系统工程,无法仅靠某个计算机软件系统来实现。这一类 workflow 系统一般也不宜采用 Lotus Notes。

在实际使用中,通常需要结合使用 Lotus Notes 和其他关系数据库系统,关系数据库中的数据经过处理后,通过 Lotus Notes 中的数据库接口(ODBC 或第三方开发的专用数据库接口软件,如 Notes Pump 等)进入 Notes 文档库,然后在 Lotus Notes 环境下发布、使用。

比较典型的适合用 Lotus Notes 开发的应用主要是办公自动化方面的应用,如公文审批及管理、计划安排、合同管理、设备管理、客户信息追踪、信息发布、分组讨论等。而银行票据处理、保险公司保单处理、图书馆管理等,则是典型的不适合用 Lotus Notes 开发的应用系统。

参考文献

- [1] 周九耀,《数据库管理系统实现技术》,中国地质大学出版社,1993
- [2] 萨师俊、王珊,《数据库系统概论》,高等教育出版社,1991
- [3] 石树刚、郑振楣,《关系数据库》,清华大学出版社,1995
- [4] Adrian Tang & Sophia Scoggins, OPEN NETWORKING WITH OSI, Prentice Hall Inc., 1992
- [5] David Gewirtz, Lotus Notes 3 Released, Art & Licensing International Inc., 1994
- [6] Lotus Notes Release 4.1 Chinese Version Online Help, 1996
- [7] Lotus Workflow Strategy, 1997
- [8] Lotus Notes R4 Overview White Paper, 1996
- [9] Office Imaging—Document Imaging in the Workgroup Environment